



# THOR: A NEW AND FLEXIBLE GLOBAL CIRCULATION MODEL TO EXPLORE PLANETARY ATMOSPHERES

JOÃO M. MENDONÇA<sup>1</sup>, SIMON L. GRIMM<sup>1,2</sup>, LUC GROSHEINTZ<sup>1</sup>, AND KEVIN HENG<sup>1</sup>

<sup>1</sup> University of Bern, Center for Space and Habitability, Sidlerstrasse 5, CH-3012, Bern, Switzerland; [joao.mendonca@csh.unibe.ch](mailto:joao.mendonca@csh.unibe.ch), [kevin.heng@csh.unibe.ch](mailto:kevin.heng@csh.unibe.ch)

<sup>2</sup> University of Zürich, Institute for Computational Science, Winterthurerstrasse 190, CH-8057, Zürich, Switzerland

Received 2016 May 11; revised 2016 July 18; accepted 2016 July 18; published 2016 September 28

## ABSTRACT

We have designed and developed, from scratch, a global circulation model (GCM) named THOR that solves the three-dimensional nonhydrostatic Euler equations. Our general approach lifts the commonly used assumptions of a shallow atmosphere and hydrostatic equilibrium. We solve the “pole problem” (where converging meridians on a sphere lead to increasingly smaller time steps near the poles) by implementing an icosahedral grid. Irregularities in the grid, which lead to grid imprinting, are smoothed using the “spring dynamics” technique. We validate our implementation of spring dynamics by examining calculations of the divergence and gradient of test functions. To prevent the computational time step from being bottlenecked by having to resolve sound waves, we implement a split-explicit method together with a horizontally explicit and vertically implicit integration. We validate our GCM by reproducing the Earth and hot-Jupiter-like benchmark tests. THOR was designed to run on graphics processing units (GPUs), which allows for physics modules (radiative transfer, clouds, chemistry) to be added in the future, and is part of the open-source Exoclims Simulation Platform ([www.exoclime.org](http://www.exoclime.org)).

**Key words:** planets and satellites: atmospheres – planets and satellites: gaseous planets – planets and satellites: individual (hot Jupiters) – planets and satellites: terrestrial planets

## 1. INTRODUCTION

The observational exploration of planetary atmospheres has shown that planets present a large diversity of climates and atmospheric circulations. This diversity raises the need to build versatile numerical tools capable of interpreting the observational data and help unveil the main mechanisms driving the atmospheric climate and dynamics. These tools have to be self-consistent and based on a theory that does not compromise the accuracy of the results under particular planetary conditions. For this purpose, our goal is to develop the first robust global circulation model (GCM)<sup>3</sup> capable of simulating a vast range of planetary conditions, which will work as a virtual “planetary lab.”

GCMs solve the complex physical and dynamical equations that include a representation of the evolution of the resolved fluid flow and various idealizations for radiative transfer, for dry or moist convection, and for heat and momentum’s turbulent surface fluxes. These models are powerful tools to simulate self-consistently, for example, the dynamical heat transport in the atmosphere and represent three-dimensional (3D) temperature maps of the atmosphere essential to interpreting observational data. They have been important to the study of the atmospheric circulation and climate of Earth, of the solar system planets, and more recently of extrasolar planets. In general, these planetary atmospheric simulations have been explored by changing the general parameters that characterize the planets (e.g., rotation rate, distance to the star, planetary radius, atmospheric mass, and composition) and adapting the physics package to deal with a wider range of atmospheric conditions (sometimes radically different from Earth). Some of the most successful simulations of the solar system planetary atmospheres are Venus (Lebonnois et al. 2010 and Mendonça & Read 2016), Mars (Forget et al. 1999), Titan (Hourdin et al. 1995), and giant gas planets like Jupiter,

Saturn, Neptune, and Uranus (Dowling et al. 1998; Yamazaki et al. 2004 and Schneider & Liu 2009). Good reviews on recent advances in Earth GCMs can be found in Randall et al. (2007) and Dowling (2013). The dynamical part of these atmospheric models usually remains the same. However, the core has been modified in Lebonnois et al. (2010) and Mendonça & Read (2016) to include, for example, variations of the specific heat with temperature for the Venus atmosphere. This numerical exploration of planetary atmospheres also expands to the extrasolar planets, as in, for example, the works on tidally locked hot Jupiters (e.g., Showman & Guillot 2002; Showman et al. 2009; Heng et al. 2011b; Kataria et al. 2015). These models obtained a robust equatorial jet feature in the simulations, which seems to be a consistent phenomenon with the observational shift of the maximum flux in the secondary eclipse seen in, for example, Knutson et al. (2009). A good review of the models that have been used for extrasolar planets can be found in Heng & Showman (2015). The majority of the 3D atmospheric models for planetary studies used as their basis dynamical cores that were developed to do Earth climate studies, with the exception of a model called EPIC (Dowling et al. 1998). The EPIC model is the only 3D climate model that we are aware of that has been developed from the ground up with the main goal of exploring different planet atmospheres.

Despite some success in planetary studies, the vast diversity of planetary characteristics observed raises questions about the flexibility of current atmospheric models to represent accurately the atmospheric physics of those planets. Planetary climate models adapted from Earth climate studies have usually included approximations that are Earth-centric. Some of the most commonly used approximations are the shallow atmosphere and the hydrostatic approximation (described in Section 3). In the new platform that we present in this work, we want to avoid making any ad hoc assumption that can compromise the physics of the problem. The amount of detailed information about extrasolar planetary atmospheres is

<sup>3</sup> Also termed “general circulation models.”

still very limited and not enough to validate 3D climate models. Because of these limitations, the modeling of extrasolar planets has to be formulated from first principles. The perfect simulation would be based on universal physical schemes with no artificial forcing that could reproduce all the observational data or help in preparing the observations (e.g., by making predictions). Those ideal “planetary labs” still do not exist, and we are still in the important process of assessing the robustness of our models for different atmospheric conditions. A careful exploration of the parameter space needed for this process can teach us important lessons on how the atmospheric circulation and climate work. In addition, it continues to be poorly understood how new atmospheric circulations are driven for a large range of astronomical and planetary bulk parameters. Numerically, we also need to improve our knowledge of the balance between physical and numerical sources and sinks of quantities such as angular momentum and total energy for a large diversity of planetary conditions, which can be crucial in assessing the accuracy and robustness of the simulations.

The dynamical core is the part of the atmospheric models that solves the resolved dynamical fluid equations, including thermodynamics and mass conservation. The new and flexible dynamical model THOR is part of the Exoclim Simulation Platform (ESP), and the code is intended to be freely open source (see more information at [www.exoclime.org](http://www.exoclime.org)). The ESP is divided into two cores: the core that solves the physics equations such as radiative transfer and convection, and the core that solves the fluid equations. The physics core used in this work is very simplified, and more sophisticated schemes are being developed to be included in the physics core. Currently, two physical schemes are being implemented: HELIOS and VULCAN. HELIOS is an efficient and flexible radiative transfer code suitable for 3D climate models (M. Malik et al. 2016, in preparation). This code will represent the radiative emission, absorption, and scattering in the atmosphere that are due to gas molecules and clouds in the ESP simulations. The scheme uses as input *k*-distribution tables, which are produced by the open-source program called HELIOS-K (Grimm & Heng 2015). Other physics modules such as VULCAN will represent the atmospheric chemistry, including thermochemistry and photochemistry (S.-M. Tsai et al. 2016, in preparation). In addition to the dynamical core presented here, based on a horizontally explicit and vertically implicit (HE-VI) type time integrator, we are also working on a fully explicit Riemann solver (L. Grosheintz et al. 2016, in preparation).

The dynamical core THOR was planned to be flexible, simple to use, and suitable for parallel computations, which will allow us to run the efficient high-resolution simulations necessary to resolve important instabilities and to improve numerical accuracy. The main goal when developing THOR was to provide a solid dynamical core capable of simulating and helping us to have a better understanding of the atmospheric dynamics and climate of a large diversity of planets.

In this work, we present the first version of the 3D atmospheric model THOR. In the next section, we will start describing the spherical grid used to discretize the model domain. In this section, we also describe the necessary modifications to the icosahedral grid to improve the numerical accuracy of the divergence and gradient operators. In Section 3, we present the equations solved by THOR and the methods used for the time and space integrations. We also discuss the main

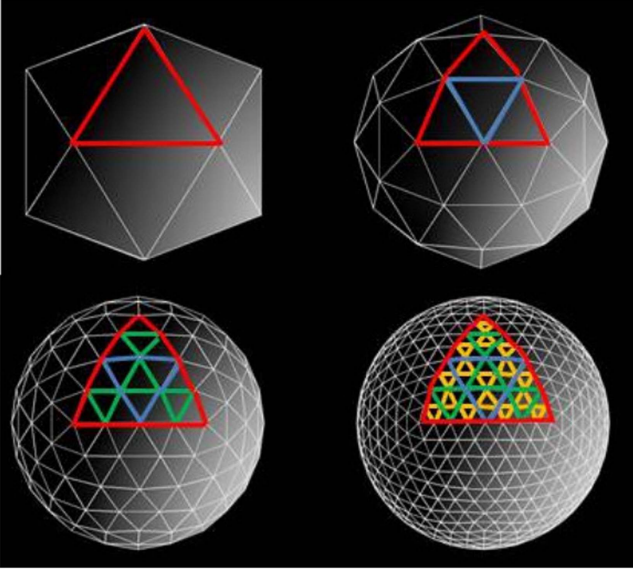
approximations often used in climate models and their validity. The numerical diffusion applied to the prognostic variables (these are variables predicted by the integration of the equations) in THOR is presented and discussed in Section 4. In Section 5, we include the description of the simple physics representations for radiation, convection, and boundary-layer friction that were coupled with the new dynamical core. In Section 6, we briefly describe the graphics processing unit (GPU) implementation of the new code. In Section 7, the new model is validated against other models for two distinct atmospheric conditions. Finally, in Section 8, we present the general conclusions.

## 2. GRID STRUCTURE

A simple latitude–longitude grid is associated with the convergence of the meridians at the poles that largely constrains the time step at high latitudes to maintain the model stability (known as the “pole problem”). In order to relax the Courant–Friedrichs–Lewy (CFL) condition at high latitudes, the equations in THOR are solved in an icosahedral grid, which is a quasi-uniform grid. This polyhedron was chosen instead of other grids that also solve the pole problem because of its higher uniformity and isotropy. The icosahedral grid was first applied to numerical atmospheric models in the 1960s by Sadourny et al. (1968) and Williamson (1968). In the beginning, these methods were not actively pursued because of the rise of other more popular methods at that time, such as spectral models. The spectral models, in general, are numerically more accurate than grid-point models. However, at high resolutions, they become computationally more inefficient than the grid-point models. Another drawback is the poor representation of large gradients or discontinuities in the spectral models, which can result in spurious waves (this is also known as “spectral ringing”). For these reasons, models based on icosahedral grids have recently become more popular (e.g., Majewski et al. 2002; Tomita & Satoh 2004 and Zängl et al. 2015), and it is the option chosen to be the core of THOR. In the next two subsections, we explain how the icosahedral grid is built and modified to improve numerical accuracy and how the main divergence and gradient are defined in the control volumes. A good review of horizontal grids for GCMs can be found in Staniforth & Thuburn (2012).

### 2.1. Standard Icosahedral

The construction of our grid starts always from the platonic solid known as the icosahedron. This solid has 20 faces (equilateral triangles), 30 edges, and 12 vertices uniformly distributed over the spherical surface. Two of the 12 vertices are placed at each pole. The icosahedral grid can also be decomposed easily into 10 rhombuses from the 20 equilateral spherical triangles, which can be handy for parallel computing (e.g., Heikes et al. 2013 and the work here). There are no uniform grids (platonic solids) with more than 12 vertices, but two methods have been used to increase the number of vertices from the initial icosahedron and keep the distribution of points quasi-uniformly distributed. The two methods are based on recursive (e.g., Baumgardner & Frederickson 1985 and Heikes & Randall 1995) and nonrecursive (e.g., Sadourny et al. 1968 and Williamson 1968) algorithms. The nonrecursive method is easier to implement than the recursive method and consists of dividing each of the 20 equilateral triangles of the original



**Figure 1.** Grid refinement using a recursive method, which consists of bisecting each edge and projecting the new points on the sphere.

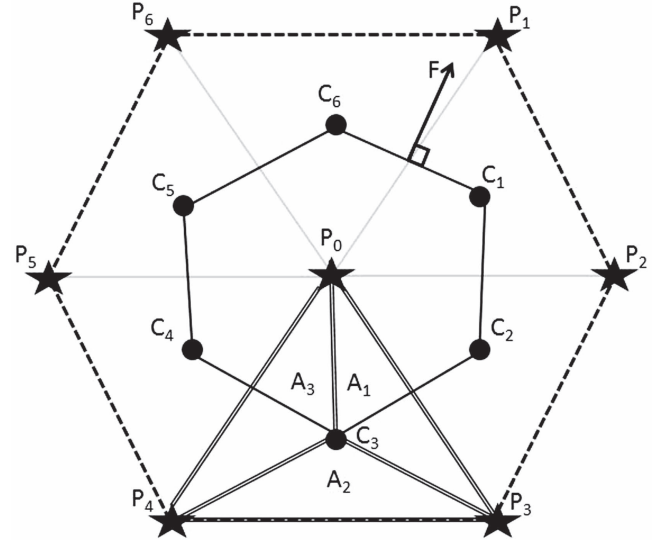
icosahedron into smaller equal sections (equilateral triangles). The intersection points from the construction of the new triangles are then projected onto the spherical surface, and they represent the new vertices of the grid. One of the main advantages of this method over the recursive method is the larger flexibility in the model resolution. However, the ratio of the longest and shortest distances between adjacent points is higher than in the recursive method (Wang & Lee 2011). This disadvantage of the nonrecursive method makes it less attractive, which can compromise the numerical accuracy, and for this reason we use the recursive algorithm in THOR. Figure 1 shows schematically how the recursive method works. In this method, the grid refinement is done by bisecting the edges of the triangles, which creates four smaller triangles. The new vertices are then projected onto the spherical surface, and the procedure is repeated until the desired resolution is reached. In this work, we follow Tomita et al. (2001) to define the level of refinement in the grid, where for example g-level 5 means that the grid was refined five times. The geometrical differences between this method and the nonrecursive one lie in the fact that when a spherical equilateral triangle is refined using the recursive method the product is three spherical isosceles triangles and one spherical equilateral triangle, and the refinement of a spherical isosceles triangle results in two isosceles triangles and two general spherical triangles (Wang & Lee 2011). At the end of the recursive method, the grid has three types of spherical triangles, and the nonrecursive has just one. A review of the geometric properties of the icosahedral grid can be found in Wang & Lee (2011).

Some important quantities that help in measuring the grid resolution are defined as

$$N_{\text{vertices}} = 2 + 10 \times 2^{2 \times \text{glevel}}, \quad (1)$$

$$\bar{d} = \sqrt{\frac{2\pi}{5}} \frac{A}{2^{\text{glevel}}}, \quad (2)$$

$$\bar{\theta} = \sqrt{\frac{2\pi}{5}} \frac{1}{2^{\text{glevel}}}, \quad (3)$$



**Figure 2.** Scheme that shows how the grid is discretized for the finite-volume method. The  $P$  points refer to the icosahedral vertices and the  $C$  points are at the center of the triangles, and they define the control volume vertices. The areas  $A_1$ ,  $A_2$ , and  $A_3$  are used to interpolate the values defined in the  $P$  points to the  $C$  points (see Equation (5)). The  $F$  vectors represent the numerical outward fluxes that cross the edges defined by the  $C$  points.

where  $N_{\text{vertices}}$  is the number of grid points in one layer,  $A$  represents the radius of the planet, and  $\bar{d}$  and  $\bar{\theta}$  are the averaged resolution in space and angle. For example, using the Jupiter radius ( $A \approx 70,000$  km) as a parameter in the model and a g-level 5 refinement of the grid results in a  $\bar{d} \approx 2452$  km ( $\bar{\theta} \approx 2^\circ$ ).

The icosahedral grid is covered by triangles, as shown in Figure 1. This grid can also be described by its dual, which is defined by connecting the centers of the triangles. The dual contains pentagons and hexagons for g-level  $> 0$ . The pentagons are centered at each point of the original icosahedron and the hexagons in the other vertices. These new geometrical forms represent the control volumes in our finite-volume method, as shown in Figure 2. We follow the method described in Tomita et al. (2001) to discretize the grid and formulate the finite-volume method. The flux at the edges of the control volume needed to calculate the divergence is defined as

$$F_k = l_k \frac{\Phi_k^* + \Phi_{k+1}^*}{2} \cdot \hat{n}_k, \quad (4)$$

where  $l_k$  is the geodesic arc length between the control volume vertices, and  $\hat{n}_k$  is the unit outward vector normal to the geodesic arc. The asterisks in the vectors indicate that the vectors are defined at the vertices  $C_k$  of the control volume (see Figure 2). The index  $k$  can take a number between one and five or six, which correspond to each edge of the pentagon or hexagon, respectively. To calculate the vectors at these positions, we use the following equation:

$$\Phi_k^* = \frac{A_2 \Phi_0 + A_3 \Phi_k + A_1 \Phi_{k+1}}{A_1 + A_2 + A_3}, \quad (5)$$

where the vectors with no asterisk are defined at  $P_k$ , the cell centers. The area of the spherical triangle  $A_1$  is defined by the points  $P_0$ ,  $P_k$ , and  $C_k$ ;  $A_2$  by the points  $P_k$ ,  $P_{k+1}$ , and  $C_k$ ; and  $A_3$  by the points  $P_0$ ,  $P_{k+1}$ , and  $C_k$  (see Figure 2). With



Equations (4) and (5) we have all the ingredients to calculate the horizontal divergence operator of a vector  $\Phi$  at point  $P_0$  using the Gauss theorem:

$$\nabla_h \cdot \Phi_0 \simeq \frac{1}{A_c} \sum_{k=1}^{6(5)} F_k, \quad (6)$$

where  $A_c$  is the area of the control volume. The sum can be expanded to five or six terms depending on the number of sides of the control volume, which can be a pentagon (five) or a hexagon (six). The horizontal gradient operator of a scalar  $s$  is calculated using

$$\nabla_h s_0 \simeq \frac{1}{A_c} \sum_{k=1}^{6(5)} l_k \frac{s_k^* + s_{k+1}^*}{2} \hat{n}_k - \frac{s_0}{A_c} \sum_{k=1}^{6(5)} l_k \hat{n}_k. \quad (7)$$

The second term in Equation (7) is a correction term that is due to the spherical space curvature. Without this term, a gradient of a constant field would not be zero.

The vertical discretization is based on layered vertical columns that are centered at points  $P$  and bounded by the control volume edges. In the numerical scheme, it is often necessary to estimate physical quantities at the interfaces between two consecutive layers or to estimate the vertical momentum at the center of the layer (the vertical momentum is updated at the interfaces). To solve these problems we use an interpolation method. The model has implemented a Lagrangian polynomial interpolation algorithm based on

$$\Phi(z_h) \approx \sum_{j=0}^k \Phi_j L_j(z_h), \quad (8)$$

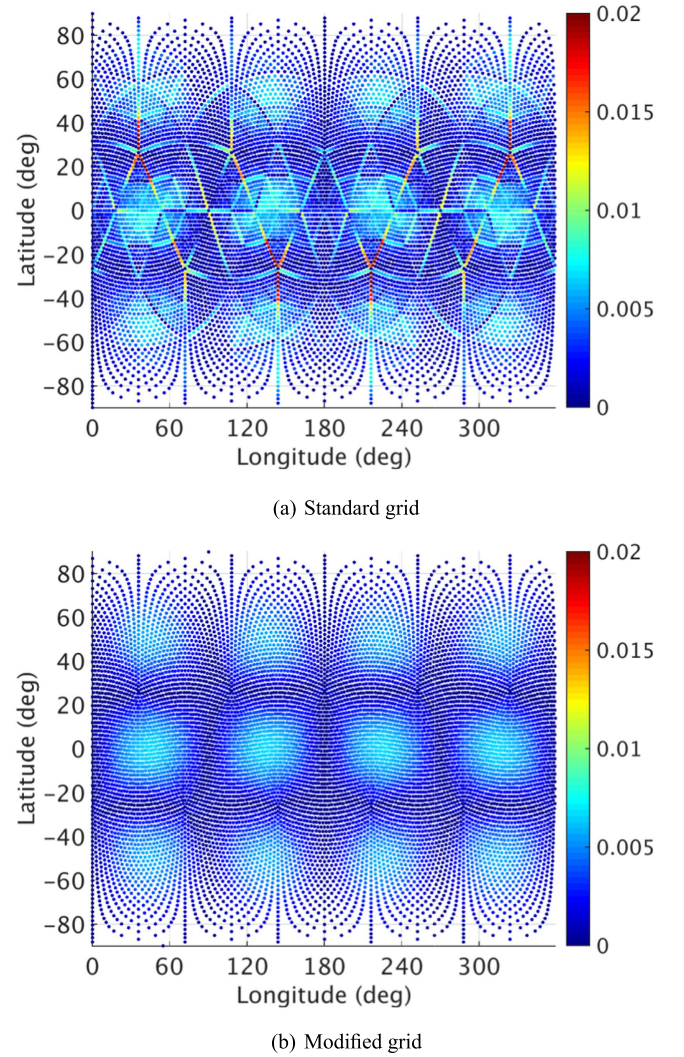
$$L_j(z) = \prod_{\substack{0 \leq i \leq k \\ i \neq j}} \frac{z_h - z_i}{z_j - z_i}, \quad (9)$$

where  $z_i$  is the altitude at the cell center and  $z_h$  at the interface, and  $\Phi(z_h)$  is the variable  $\Phi$  interpolated to an altitude  $z_h$ . In the simulations explored later we use  $k = 1$ , which is a linear interpolation, and we did not find it necessary to use higher-order interpolations.

The accuracy of the divergence and gradient operators from Equations (6) and (7), when applied to the standard icosahedral grid described above, is low and constrains drastically the performance of the atmospheric model (see Figure 3). In the following section, we describe the modifications applied to the grid in order to improve the accuracy of the operators.

## 2.2. Modified Icosahedral Grid

Numerical noise from the space integration can be expected to reveal the underlying grid structure in the numerical solution, which is associated with a numerical problem called grid imprinting. The icosahedral grid in this section is modified to increase the numerical accuracy of the divergence and gradient operators. The poor accuracy is related to the nonuniform distribution of the triangular areas. As we described above, the division of the triangles during the grid refinement results in triangles with different areas, and in Tomita et al. (2001) it is argued that regions with the largest area gradient correspond to regions with higher grid noise. The larger distortions in the grid are located adjacent to the geodesics from the initial icosahedron. These distortions can also be measured



**Figure 3.** Absolute error for  $\nabla \cdot \mathbf{v}$  defined in Equation (16) when  $m = 1$  and  $n = 1$ . In map (b) we apply the “spring dynamics” method and move the center of the control volumes to the centroid.

by the ratio of the longest and shortest distances between adjacent points, as described in the previous section, and it is larger in regions of larger grid distortion. Despite the recursive method being smoother than the nonrecursive method, we still apply a method called “spring dynamics” to the grid (Tomita et al. 2001) to smooth it. After this method is applied, we do a second step that consists of moving the new icosahedral vertices to the centroid of the control volumes, which allows the model to reach an effective accuracy close to the second order. Note that the order of accuracy is related to the rate of convergence of the numerical solution to the exact solution, which means that the error in a second-order scheme decays quadratically with the resolution.

As pointed out above, the first step to improve the quality of the grid and smooth the grid distortions is to apply “spring dynamics” (Tomita et al. 2001). In this method, all of the grid points are connected to their nearest neighbors by springs. The modified grid is the icosahedral grid obtained by recursive refinements and with a radius equal to one (nondimensional radius from the original icosahedron). In the “spring dynamics” method, the net force applied to each grid point due to the

attached springs is represented by

$$\frac{d\mathbf{v}}{dt} = \sum_{i=1}^6 k(d_i - \bar{d}_{\text{spr}})\mathbf{e}_i - \alpha\mathbf{v}. \quad (10)$$

In this equation,  $\alpha$  is the friction coefficient,  $k$  is the spring constant,  $d$  is the arc length between the central point and one of the neighbor points, which in the case shown in Equation (10) is between  $P_0$  and  $P_i$ . The values of  $i$  vary between 1 and 6 because the position of the center of the 12 initial pentagons remains fixed. Also,  $\bar{d}_{\text{spr}}$  is the natural spring length, and  $\mathbf{v}$  is the velocity of the unit mass grid points. The latter is defined simply by how much the points move:

$$\mathbf{v} = \frac{d\mathbf{P}_0}{dt} \quad (11)$$

The natural spring length  $\bar{d}_{\text{spr}}$  can be written as

$$\bar{d}_{\text{spr}} = \beta \frac{2\pi A}{10 \times 2^{\text{glevel}-1}}. \quad (12)$$

The parameter  $\beta$  is a tuning parameter, and it was set to 1.15 in our work, as suggested by Tomita et al. (2001). The friction term allows the system to converge smoothly to an equilibrium, avoiding possible instabilities created by a sudden move of the grid points. The parameter  $\alpha$  that controls the friction is a tunable parameter, and we set this value to 1 for simplicity. The dynamical system is integrated with a time step of 0.01 until the net force in each point becomes very small. The simulation is stopped following the condition that the points do not move more than  $10^{-5}$  is satisfied at every point. Note that there is no need to associate the variables in this scheme with any units because of the simplicity and the geometric goal of the method. After the grid distortions were smoothed with the “spring dynamics” scheme, the grid points are moved to the centroid of the control volume to increase the accuracy of the finite-volume method (Tomita et al. 2001). In this step, we first correct the position of the vertices ( $C_k$ ) of the control volumes by moving them to the centroid of the triangles. Next, we keep the vertices fixed and move the center ( $P_i$ ) of the control volume to its new centroid. The new positions are found using the following equation:

$$\mathbf{r}_c = \frac{1}{2\pi} \sum_{i=1}^n \frac{\mathbf{r}_i \times \mathbf{r}_{i+1}}{|\mathbf{r}_i \times \mathbf{r}_{i+1}|} \tan^{-1}(|\mathbf{r}_i \cdot \mathbf{r}_{i+1}|/|\mathbf{r}_i \times \mathbf{r}_{i+1}|). \quad (13)$$

In this equation,  $\mathbf{r}_c$  is the new centroid;  $n$  can be three, five, or six, depending on if it is the centroid of a triangle, pentagon, or hexagon; and  $\mathbf{r}$  is the point position.

To validate the accuracy of the operators with a new modified grid, we used the test functions proposed by Heikes & Randall (1995):

$$\alpha(\lambda, \varphi) = \sin(\lambda), \quad (14)$$

$$\beta(\lambda, \varphi) = \cos(m\lambda) \cos^4(n\varphi), \quad (15)$$

$$\begin{aligned} \mathbf{v} &= \alpha \nabla \beta \\ &= -m \frac{\cos^4(n\varphi)}{\cos(\varphi)} \sin(\varphi) \sin(m\lambda) \mathbf{i} \\ &\quad - 4n \cos^3(n\varphi) \sin(n\varphi) \sin(\lambda) \cos(m\lambda) \mathbf{j}. \end{aligned} \quad (16)$$

The quantities that we will test are  $\nabla\beta$  and  $\nabla \cdot \mathbf{v}$  for the solutions when  $m = 1$  and  $n = 1$ , and when  $m = 3$  and  $n = 3$ . These tests challenge the accuracy of the operators across the sphere, and the accuracy can be quantified because the exact solution is known. To quantify the error in the operators, we define the identities

$$l_2 = I[(\phi(\lambda, \varphi) - \phi_{\text{exact}}(\lambda, \varphi))^2]^{1/2}, \quad (17)$$

$$l_\infty = \max|\phi - \phi_{\text{exact}}|, \quad (18)$$

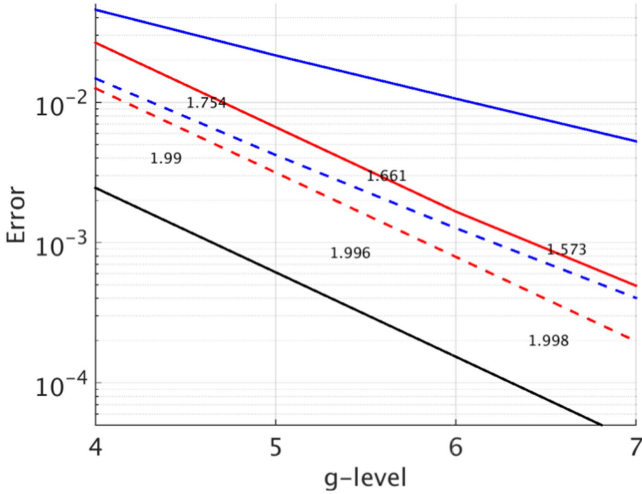
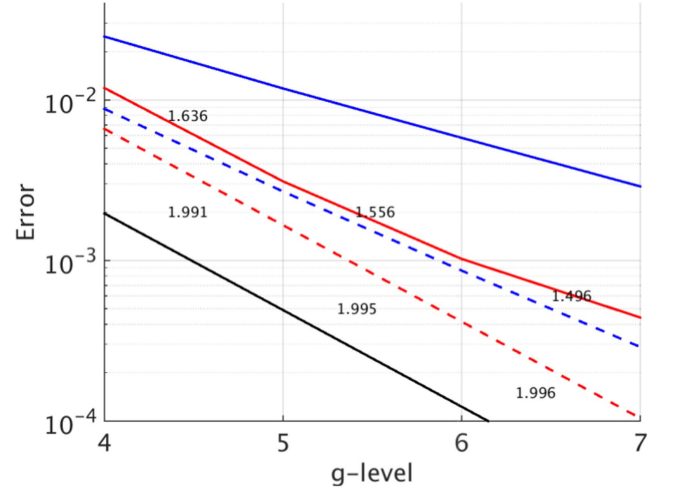
where the function  $\phi$  can be  $\nabla \cdot \mathbf{v}$  or  $\nabla\beta$ , and the function  $I$  represents the global average.

As a first step in the error analysis, we compute the absolute error of  $\nabla \cdot \mathbf{v}$  for the solution when  $m = 1$  and  $n = 1$ . Figure 3 shows the distribution of the errors in the modified and unmodified grids. As we mentioned before, the largest errors are located in the regions with the largest grid distortions, which are located adjacent to the geodesic that represents the initial icosahedron. When the “spring dynamics” method is applied and the centers of the control volumes are moved to the centroids, the grid noise is efficiently removed. The largest amplitudes in the modified grid case are then related to the maxima of the function  $\nabla \cdot \mathbf{v}$ .

Now that we have gained more intuition on the location of the largest errors, we explore the accuracy of the divergence and gradient operators using  $l_2$  and  $l_\infty$ . Our goal when we modified the icosahedral grid was to reach an order of accuracy close to the second order, which is the theoretical maximum order for the finite-volume scheme with a linear spatial reconstruction (LeVeque 2002). Here,  $l_2$  represents how on average the root mean square error is converging, and in the case of  $l_\infty$  it is associated with the convergence of the point with the least accuracy. Figures 4 and 5 show the convergence plots. In Figure 4, we analyze the accuracy of the model when computing  $\nabla \cdot \mathbf{v}$  for different grid resolutions. For both test functions, the grid modification clearly improved the accuracy (Figures 4 and 5). This is also reflected in improved  $l_2$  convergence rates where we reach second order for the modified grid. The  $l_\infty$  case is similar to  $l_2$ ; however, in the solution with  $m = 1$  and  $n = 1$ , the last slope is slightly smaller than the  $l_2$  case. The analysis of the results in Figure 5 for the quantity  $\nabla\beta$  is very similar to Figure 4.

In summary, the modifications to the grid improve the accuracy of the divergence and gradient operators. Despite in some cases the convergence of  $l_\infty$  being slower than the globally averaged case, an accuracy close to second order is reached in all grid points. Later in Section 7.1, we also verify that in a long-term simulation the grid noise is not detected in the model results and has no impact on the flow simulated.

There are different ways of storing the physical quantities in the grid. In our current model, we use the Arakawa-A grid, where all the physical variables are defined at the center of the control volume. This structure has been tested in models that use icosahedral grids, such as Stuhne & Peltier (1996) and Tomita et al. (2001). This decision is associated with the set of dynamical equations that we solve in THOR (presented in the next section) and the numerical scheme described above. Our strategy was to use a simple and efficient solver. Other options have been proposed in the literature for icosahedral grids, such as the ZM-grid in Ringler & Randall (2002). In the ZM-grid the velocity components and the mass are defined at different positions (staggered grid). This grid allows instabilities related

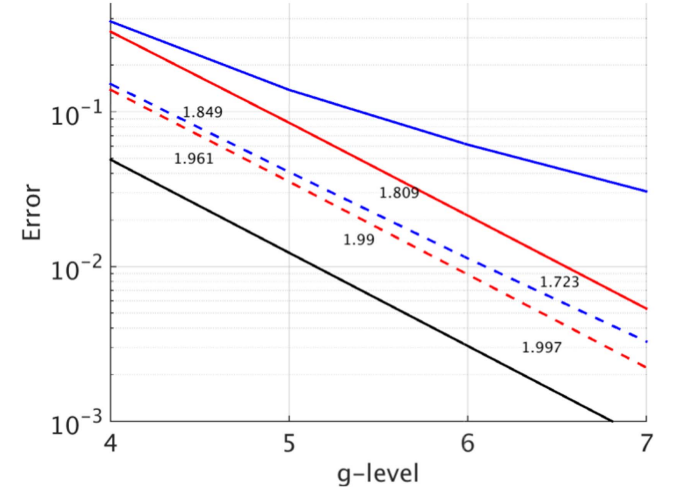
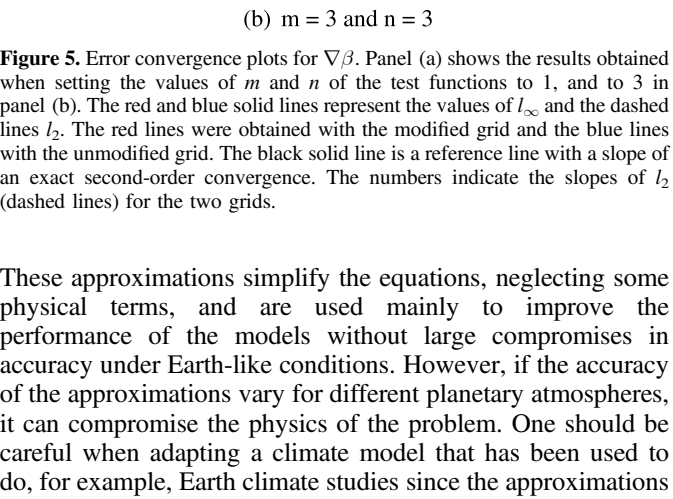
(a)  $m = 1$  and  $n = 1$ (b)  $m = 3$  and  $n = 3$ 

**Figure 4.** Error convergence plots for  $\nabla \cdot \mathbf{v}$ . Panel (a) shows the results obtained when setting the values of  $m$  and  $n$  of the test functions to 1, and to 3 in panel (b). The red and blue solid lines represent the values of  $l_\infty$  and the dashed lines  $l_2$ . The red lines were obtained with the modified grid and the blue lines with the unmodified grid. The black solid line is a reference line with a slope of an exact second-order convergence. The numbers indicate the slopes of  $l_2$  (dashed lines) for the two grids.

to the fact that the velocity has higher resolution than the mass to grow (these instabilities are also known by computational modes). However, if the computational modes are efficiently filtered, the ZM-grid has advantages over all the other grids in the literature, such as a better representation of the geostrophic adjustment (Ringler & Randall 2002). We have to use a higher horizontal resolution in THOR to reach the same accuracy as models that use ZM-grids.

### 3. DYNAMICAL EQUATIONS AND APPROXIMATIONS

THOR represents resolved fluid dynamical phenomena in the atmosphere. The new model solves the 3D nonhydrostatic compressible Euler equations on a rotating sphere. Our goal was to reduce as much as possible the number of approximations usually applied to the Euler equations in Earth climate models that would compromise the flexibility of the model.

(a)  $m = 1$  and  $n = 1$ (b)  $m = 3$  and  $n = 3$ 

**Figure 5.** Error convergence plots for  $\nabla\beta$ . Panel (a) shows the results obtained when setting the values of  $m$  and  $n$  of the test functions to 1, and to 3 in panel (b). The red and blue solid lines represent the values of  $l_\infty$  and the dashed lines  $l_2$ . The red lines were obtained with the modified grid and the blue lines with the unmodified grid. The black solid line is a reference line with a slope of an exact second-order convergence. The numbers indicate the slopes of  $l_2$  (dashed lines) for the two grids.

These approximations simplify the equations, neglecting some physical terms, and are used mainly to improve the performance of the models without large compromises in accuracy under Earth-like conditions. However, if the accuracy of the approximations vary for different planetary atmospheres, it can compromise the physics of the problem. One should be careful when adapting a climate model that has been used to do, for example, Earth climate studies since the approximations included in the model can be very Earth-centric. Some of the most often used approximations that we *avoid* are as follows:

1. **Hydrostatic approximation:** The vertical pressure gradient balances the gravitational force. This approximation implies that the vertical scales of the atmospheric motions are smaller than the horizontal scales so that  $H/L \ll 1$ , where  $H$  is the scale height and  $L$  the characteristic horizontal scale. The hydrostatic assumption eliminates the vertical propagation of acoustic waves because the



acoustic terms are neglected. If the model is able to resolve important atmospheric phenomena with horizontal scales smaller than the scale height (e.g., cumulus convection or fine-scale gravity waves), then this approximation can compromise the accuracy of the results.

2. Shallow atmosphere approximation: The spherical components of the fluid equations that depend on  $1/r$  are changed to the constant  $1/A$ , where  $A$  is the mean planetary radius. This approximation simplifies the equations, which is justified if the atmosphere explored is much thinner than the distance from the center of the planet to the region simulated. However, this approximation has extra costs, as pointed out in Phillips (1966). In order to conserve axial angular momentum, some metric terms have to be neglected, as are the Coriolis terms involved in the horizontal components of the planet's angular velocity. These approximations are known as the “traditional approximations.”
3. Anelastic approximation: The density in the divergence term of the mass continuity equation is replaced by a reference density that only depends on altitude. The original formulation for the anelastic approximation was suggested by Ogura & Phillips (1962). The idea of this approximation was to neglect the elasticity of the fluid and consequently filter sound waves. This approximation allows larger steps in the time integration than in the integration of the full compressible model without assuming hydrostatic balance. This approximation assumes that the physical thermodynamic quantities have just small variations from the basic state.

The approximations described above are not used in this work, but they have been implemented in the model, and the user can switch them on and off to explore their impact on the results. Despite not using any of these approximations, we still make some assumptions about the atmospheres being simulated, such as the atmosphere not being ionized and the effective gravity being constant and radial. In the future, we will develop modules that will complement and improve the physical representation of these two cases. The first case becomes important in highly irradiated or very hot atmospheres and requires a 3D magnetohydrodynamic treatment of the equations (e.g., Rogers & Showman 2014). The assumption for constant effective gravity breaks down if we are exploring deep atmospheres. The radial constant gravity assumption can be easily fixed in the equations, but it is a good approximation of the two test atmospheres explored in this work. In fast-rotating planets, such as Jupiter, where the centrifugal forces ( $-\Omega \times (\Omega \times \mathbf{r})$ ) are not negligible compared to the Newtonian gravitational forces ( $-\nabla\Phi_N$ ), spherical geopotential surfaces become a poor approximation to represent the geopotential surfaces (White et al. 2005). In this case, the apparent gravity is defined as

$$\mathbf{g} \equiv -\nabla\Phi = -\nabla\left(\Phi_N - \frac{\Omega^2 s^2}{2}\right), \quad (19)$$

where  $s$  is the perpendicular distance to the rotation axis. Since the apparent gravity is a dominant force, it is convenient to express the fluid equations in a spheroidal coordinate system, as suggested in, for example, Staniforth (2014).

The equations in THOR are written in the flux form, which helps conserve our prognostic variable in the model domain. A good performance of a dynamical core requires that the timescale of spurious a numerical source or sinks of conservation quantities are longer than the physical source or sink timescales. The conservation properties of quantities that have the longest physical source or sink timescales are the ones that require more attention (Thuburn 2008). The three main equations that describe the flow in a dry atmosphere in THOR are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (20)$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) = -\nabla p - \rho g \hat{\mathbf{r}} - 2\rho \Omega \times \mathbf{v}, \quad (21)$$

$$\frac{\partial \rho \theta}{\partial t} + \nabla \cdot (\rho \theta \mathbf{v}) = 0. \quad (22)$$

The first equation represents the mass conservation, where  $\rho$  is the atmospheric density and  $\mathbf{v}$  the velocity. Mass conservation is one of the most fundamental conservation properties in climate models. Dry mass is a robust invariant in the model, and it should be conserved independently of any diffusion or friction processes acting in the atmosphere. Any spurious perturbations in the mass distribution will have an impact on the pressure field, which affects, for example, the flow motion.

Equation (21) represents Newton's second law of fluid motion. In this equation,  $p$  is the atmospheric pressure,  $\Omega$  is the planetary rotation rate,  $g$  is the gravity,  $\hat{\mathbf{r}}$  is the unit vector in the radial direction, and  $\otimes$  is the tensor product. This equation is essential in the representation of the atmospheric flow. The momentum in the atmosphere is not a robust invariant, unlike the mass described above, because the momentum is transferred to smaller scales or it can be dissipated at the boundary layer. This transfer of kinetic energy is described in Section 4. The second term on the left side of the equation is a nonlinear term called the advection term. On the right we have the balance of three forces: the first one is produced by the pressure gradient, the second is the gravitational force, and the last one is related to the planet's rotation and is called the Coriolis force.

Equation (22) is the thermodynamic equation, and it is formulated in the flux form for entropy. For this equation, we followed the work of Ooyama (1990), Skamarock & Klemp (2008), and Ullrich & Jablonowski (2012). Entropy is closely linked to potential temperature  $\theta$ , where the specific entropy  $S$  can be defined by

$$dS = C_p d(\ln \theta) \quad (23)$$

where  $C_p$  is the specific heat at constant pressure. From our set of equations, it is essential to recover the pressure that is used to solve Equation (21). The pressure is calculated from

$$p = p_{\text{ref}} \left( \frac{R_d(\rho\theta)}{p_{\text{ref}}} \right)^{\frac{C_p}{C_v}}. \quad (24)$$

In the equation above, the variable  $p_{\text{ref}}$  refers to a reference pressure level, and for the simulations tested in Section 7,  $p_{\text{ref}}$  was set to 1 bar. The variable  $R_d$  is the gas constant for dry air, and  $C_v$  is the specific heat at constant volume.

This thermodynamic equation does not include explicitly the conversion between mechanical energy and thermal energy. In a real atmosphere, there is transfer of kinetic energy from large to smaller scales, and at some point this transfer of energy crosses the truncation limit of the model. This energy is dynamically transported and eventually dissipated by frictional heating (molecular scale), and it is reintroduced into the resolved scales by internal energy. Part of this cycle happens at unresolved scales in the model. In Rauscher & Menou (2012), the dissipative kinetic energy is reintroduced directly into the energy cycle as internal energy. Another possible solution to close this cycle is to solve the total energy equation in the flux form (Tomita & Satoh 2004) instead of the entropy equation used in THOR. These solutions would close the cycle since they are conserving the total energy, but they can lead to local spurious conversions between available and unavailable potential energy, which would result in a poor performance of the model. There are also numerical schemes that inject back some of the dissipative energy into the resolved kinetic energy using a stochastic pattern generator in order to improve the atmospheric circulation at scales close to the truncation limit (Shutts 2005). However, it is not clear how the rate of energy backscatter can be generalized for planetary applications. We are still investigating the best options to represent the missing physics that can be important in atmospheres that have a large transfer of kinetic energy to unresolved scales. We have implemented and tested, in order to keep energy balance in the model, a scheme proposed by Williamson et al. (2009), which globally uniformly adjusts the internal energy at all grid points. Numerical inaccuracies in the total energy conservation can lead to drifts in the mean circulation (e.g., Boville & Gent 1998), and this scheme helps to alleviate the problem. However, because of its low impact on the results of the simulations explored in this work, the results presented later were obtained without any energy fixer scheme.

To complement the main set of equations, we use the assumption that the atmosphere can be treated as an ideal gas, which is a good approximation for the atmospheric conditions explored in this work:

$$p = \rho R_d T, \quad (25)$$

where  $T$  is the atmospheric temperature. A more general equation of state can be included in THOR, which implies two modifications: one, on how the pressure is estimated after the entropy equation is integrated, and two, on how the pressure equation is used to do the implicit vertical integration of the momentum.

The 3D divergence and gradient operators are split as

$$\nabla \Phi = \nabla_h \Phi + \hat{r} \frac{d\Phi}{dr}, \quad (26)$$

$$\nabla \cdot \Phi = \nabla_h \cdot \Phi_h + \frac{1}{r^2} \frac{d}{dr} (r^2 \Phi_r), \quad (27)$$

where  $r$  is the distance from the planet center,  $\hat{r}$  is the unit radial vector, and  $\nabla_h$  and  $\nabla_r$  are the spherical gradient and divergence operators defined in Equations (6) and (7). In the second equation, the variables  $\Phi_h$  and  $\Phi_r$  are the horizontal (parallel to the spherical surface) and radial projections of the vector  $\Phi$ . These two quantities are calculated in the model

using

$$\Phi_h = \Phi - \Phi \cdot \hat{r}, \quad (28)$$

$$\Phi_r = \Phi \cdot \hat{r}, \quad (29)$$

where  $\hat{r}$  is the unit radial vector.

THOR solves the full Euler equations that allow the development of, for example, fast acoustic waves that constrain the time step needed to satisfy the CFL condition. To overcome this restriction, we use a split-explicit method from Wicker & Skamarock (2002) and Skamarock & Klemp (2008), coupled with an HE-VI method (Tomita & Satoh 2004). The split-explicit method involves splitting the time integration of the Euler equation into short explicit steps for the terms associated with acoustic modes and large steps for the rest. The time-splitting method is simple to implement and improves the efficiency of the integration, allowing the CFL condition to be obeyed in the horizontal integration. In the vertical direction, we have typically much higher resolution than in the horizontal one, and the fast vertically propagating waves are handled with an implicit scheme. Other techniques have been explored to make nonhydrostatic models soundproof. One example is the semi-implicit method, where the horizontal and vertical propagation of sound waves is treated implicitly (Durran & Blossey 2012). However, this method requires the expensive numerical solution of a 3D elliptic equation. More recently, Dubos & Voitus (2014) extended the work from Arakawa & Konor (2009) and derived the equations for a soundproof semihydrostatic model from Hamilton's principle of least action. The new set of equations retains the vertical acceleration term, which provides accurate model results at nonhydrostatic scales and filters efficiently the acoustic waves from the atmospheric flow due to the constraint to the air parcels to stay close to their hydrostatic position. But as pointed out in Dubos & Voitus (2014), it is still not clear if the integration of the new equations is more efficient than the integration of the full Euler equations with the method used in our work (Skamarock & Klemp 2008). Note that in this work we are not interested in resolving the sound waves that are in general energetically very low and have almost no impact on the atmospheric circulation and climate. To resolve correctly the sound waves with THOR, the time step of the time integration has to be reduced until the CFL condition for these fast waves is satisfied in any direction.

Equations (20)–(22) are divided into terms related to fast and slow physical processes. This splitting, as we said above, is done to improve the performance of the time integration. The strategy is to integrate the different terms with different numerical methods: the fast terms, related to the compressible terms that generate fast acoustic waves, are integrated with a simple and fast scheme and using a short time step; and the slow terms have a more accurate scheme and longer time step. Figure 6 shows schematically how the time integration method works. The time-splitting method used in this work is similar to the ones described in Wicker & Skamarock (2002), Tomita & Satoh (2004), and Skamarock & Klemp (2008). For the integration of the slow terms, we use a time integration scheme proposed by Wicker & Skamarock (2002), which is a



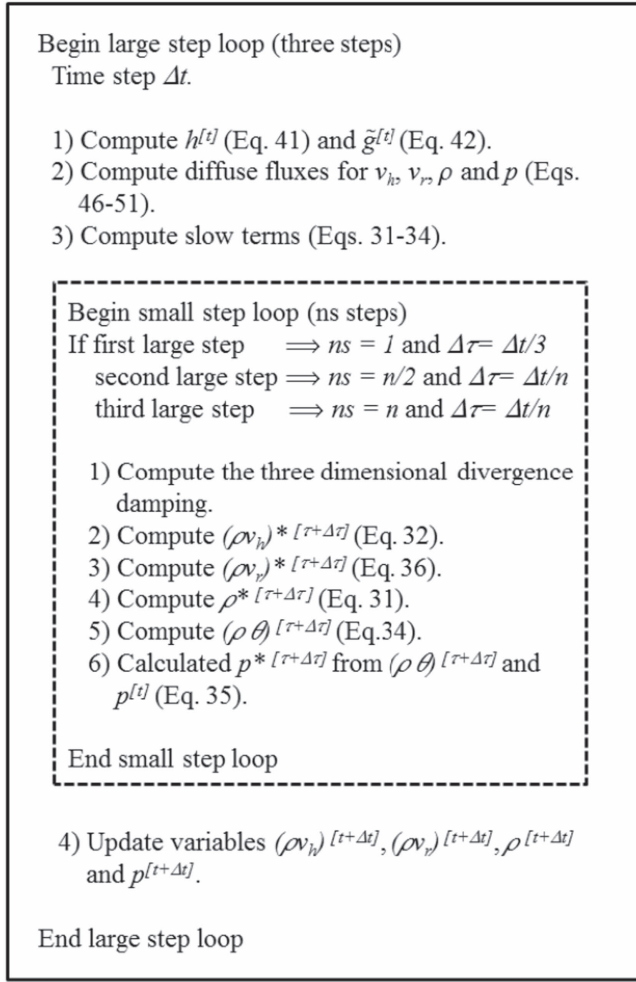


Figure 6. Time integration scheme.

predictor–corrector scheme with second-order accuracy:

$$\begin{aligned}
 \Phi^{(1)} &= \Phi^{[t]} + \frac{\Delta t}{3} R(\Phi^{[t]}) \\
 \Phi^{(2)} &= \Phi^{[t]} + \frac{\Delta t}{2} R(\Phi^{(1)}) \\
 \Phi^{[t+\Delta t]} &= \Phi^{[t]} + \Delta t R(\Phi^{(2)})
 \end{aligned} \tag{30}$$

where  $\Delta t$  is the large time step,  $\Phi$  is a prognostic variable, and  $R$  the equation terms that estimate the rate of  $\Phi$ . In the inner loop, we use a forward–backward scheme with a short time step. The number of inner loops  $n$  and also the time step of the short steps  $\Delta\tau$  change for each large time step  $\Delta t$  (Klemp et al. 2007). In the first large time step, the inner loop is just done once with a time step equal to  $\Delta t/3$ , in the second large step the inner loop runs  $n/2$  times with a time step of  $\Delta t/n$ , and in the last large step the inner loop runs for  $n$  times with a time step of  $\Delta t/n$ . A large value for  $n$  improves the stability of the model, but it becomes very time expensive. In the simulations explored later, we set the value of  $n$  to 6. To improve the accuracy of the split-explicit method, the equations are integrated using a perturbation formulation. During the inner loop, the deviations ( $\Phi^*$ ) from the large time step ( $\Phi^{[t]}$ ) are

defined as

$$\Phi^* = \Phi - \Phi^{[t]}. \tag{31}$$

Below we write the equations solved where  $\tau$  is the time step of the inner loop and  $t$  is the large time step:

$$\begin{aligned}
 &\frac{\rho^*[\tau+\Delta\tau] - \rho^*[\tau]}{\Delta\tau} + \nabla_h \cdot (\rho v_h)^*[\tau+\Delta\tau] \\
 &+ \frac{1}{r^2} \frac{\partial}{\partial r} (\rho v_r)^*[\tau+\Delta\tau] r^2 = -\nabla_h \cdot (\rho v_h)^{[t]} \\
 &- \frac{1}{r^2} \frac{\partial}{\partial r} (\rho v_r)^{[t]} r^2,
 \end{aligned} \tag{32}$$

$$\begin{aligned}
 &\frac{(\rho v_h)^*[\tau+\Delta\tau] - (\rho v_h)^*[\tau]}{\Delta\tau} + \nabla_h p^*[\tau] = -\nabla_h p^{[t]} \\
 &- \mathcal{A}_h^{[t]} - \mathcal{C}_h^{[t]},
 \end{aligned} \tag{33}$$

$$\begin{aligned}
 &\frac{(\rho v_r)^*[\tau+\Delta\tau] - (\rho v_r)^*[\tau]}{\Delta\tau} + \frac{\partial}{\partial r} p^*[\tau+\Delta\tau] \\
 &+ \rho^*[\tau+\Delta\tau] g = -\frac{\partial}{\partial r} p^{[t]} - \rho^{[t]} g - \mathcal{A}_r^{[t]} - \mathcal{C}_r^{[t]},
 \end{aligned} \tag{34}$$

$$\begin{aligned}
 &\frac{(\rho \theta)^{[t+\Delta\tau]} - (\rho \theta)^{[t]}}{\Delta\tau} + \nabla_h \cdot \theta^{[t]} (\rho v_h)^{[t+\Delta\tau]} \\
 &+ \frac{\partial}{\partial r} \theta^{[t]} (\rho v_r)^{[t+\Delta\tau]} = 0.
 \end{aligned} \tag{35}$$

The slow terms are written on the right-hand side of the equations. The variables  $\mathcal{A}$  and  $\mathcal{C}$  are the advection and Coriolis terms. These two terms are calculated using the nonprojected velocities and then projected parallel to the spherical surface and in the radial direction using Equations (28) and (29). We do not calculate the deviation of  $\theta$  from the large step directly from Equation (35) to avoid negative values that would compromise the conversion to pressure values needed for Equations (33) and (34). Instead we do the following:

$$p^*[\tau+\Delta\tau] = p_{\text{ref}} \left( \frac{R_d (\rho \theta)^{[t+\Delta\tau]}}{p_{\text{ref}}} \right)^{\frac{c_p}{c_v}} - p^{[t]}. \tag{36}$$

The first equation integrated in the model is the momentum Equation (33), which is done using an explicit method. After the horizontally projected momentum is updated, we solve the implicit Equation (34). Integrating the vertical momentum using an implicit method, we improve the performance of the model because of the instabilities created by the fast propagation of the sound waves in the typically fine vertical resolution while keeping a reasonable time step. The method used to solve Equation (34) in a nonhydrostatic framework is called the horizontal explicit and vertical implicit scheme, and it was proposed in Satoh (2002) and Tomita & Satoh (2004). Equations (32) and (35) are arranged for  $\rho^*[\tau+\Delta\tau]$  and  $p^*[\tau+\Delta\tau]$ , and the two solutions are inserted into Equation (34).

The new equation for the vertical momentum is

$$\begin{aligned}
& -\frac{1}{r^2} \frac{\partial^2}{\partial r^2} r^2 h^{[t]} (\rho v_r)^{*[\tau+\Delta\tau]} + \frac{2}{r^3} \frac{\partial}{\partial r} r^2 h^{[t]} (\rho v_r)^{*[\tau+\Delta\tau]} \\
& - \frac{\partial}{\partial r} \tilde{g}^{[t]} (\rho v_r)^{*[\tau+\Delta\tau]} - \frac{C_v}{R_d} \frac{g}{r^2} \frac{\partial}{\partial r} r^2 (\rho v_r)^{*[\tau+\Delta\tau]} \\
& + \frac{C_v}{R_d \Delta\tau^2} (\rho v_r)^{*[\tau+\Delta\tau]} = \frac{C_v}{R_d} \left[ \frac{1}{\Delta\tau^2} (\rho v_r)^{*[\tau]} \right. \\
& \left. - \frac{\partial}{\partial r} S_p - \frac{1}{\Delta\tau} \frac{\partial}{\partial r} p^{[t]} - \frac{g}{\Delta\tau} (S_p + \rho^{[t]}) + \frac{S_{v_r}}{\Delta\tau} \right], \quad (37)
\end{aligned}$$

where  $h$  is the enthalpy and  $\tilde{g}$  is the effective gravity defined by

$$h^{[t]} = C_p T^{[t]}, \quad (38)$$

$$\tilde{g}^{[t]} = \frac{1}{\rho^{[t]}} \frac{\partial p^{[t]}}{\partial r}. \quad (39)$$

The functions  $S_p$ ,  $S_\rho$ , and  $S_{v_r}$  needed to calculate  $C_0$  are computed from

$$\begin{aligned}
S_p = & -\frac{R_d}{C_v} (\nabla_h \cdot (h^{[t]} (\rho v_h)^{*[\tau+\Delta\tau]}) - \nabla_h \cdot (h^{[t]} (\rho v_h)^{[t]})) \\
& - \frac{1}{r^2} \frac{\partial}{\partial r} (h^{[t]} (\rho v_r)^{[t]} r^2) + v_h^{[t]} \cdot \nabla p^{[t]} - \tilde{g}^{[t]} (\rho v_r)^{[t]} \Big), \quad (40)
\end{aligned}$$

$$S_p = -\nabla_h \cdot (\rho v_h)^{*[\tau+\Delta\tau]} - \nabla_h \cdot (\rho v_h)^{[t]} - \frac{1}{r^2} \frac{\partial}{\partial r} (\rho v_r)^{[t]} r^2, \quad (41)$$

$$S_{v_r} = -\frac{\partial}{\partial r} p^{[t]} - \rho^{[t]} g - A_r^{[t]} - C_r^{[t]}. \quad (42)$$

After all the coefficients are calculated, we use a staggered grid to solve Equation (37). This equation is discretized in a finite-difference form over intervals of  $\Delta z$  (thickness of the layers), written as a tridiagonal matrix, and solved numerically using a Thomas algorithm. Using this method, the vertical momentum is updated at the model layer interfaces. This solver is applied to every vertical column in the grid. The solution requires two boundary conditions at the top and bottom of the model domain. In our current version of THOR, we set the vertical velocities to zero at the boundaries. As shown in Staniforth et al. (2003), the only way to formally enforce total energy and mass conservation in simulations is by applying a rigid lid on time and space at the boundaries.

After the vertical momentum equation is solved, we update the new densities from Equation (32). This equation is calculated using an implicit formulation and the new updated momentum values. Finally we integrate Equation (35) to compute the new updated potential temperatures.

#### 4. NUMERICAL DISSIPATION

Numerical dissipation schemes are an important part of atmospheric models. These routines have an important role keeping the model numerically stable, and every model needs a representation of numerical dissipation, which can be associated with inherent dissipation from the numerical methods used or represented explicitly in the numerical scheme. These schemes are intended to alleviate the accumulation of potential enstrophy (enstrophy quantifies the intensity of the vorticity) at the smallest grid scales, which could be produced by a physical

downscale cascade, numerical noise from the space and time integration, misrepresentation of nonlinear interactions (aliasing effect), or poor initial conditions. Physical interpretations of the diffusion schemes are often poor or absent, and one of the main efforts when developing these schemes is to link them to simple representations of turbulence and eddy viscosity in the subgrid scale.

The strength of the numerical dissipation is associated with the damping timescales of the waves present in the simulation. This quantity can have an important impact on large-scale phenomena, and in the case of inappropriately strong diffusion it can make fine structures disappear and affect the wave-mean flow interaction in the atmosphere, but at the same time it needs to ensure an ideal energy dissipation rate at the truncation grid scales (e.g., Thrastarson & Cho 2011). In Heng et al. (2011b), it is shown that the calibration of the numerical horizontal dissipation can have an important impact on the estimation of the temperature and wind magnitude values under typical hot Jupiter conditions. Our current model uses an explicit dissipation representation, and the strength of the diffusion is a tunable parameter, which was chosen to be large enough to ensure a stable model integration, and it does not change with latitude or longitude. Currently, we are also investigating other possibilities for the diffusion strength, such as the ones based on nonlinear formulations (Smagorinsky 1963). Another issue that is being studied is the impact of a formulation of the numerical dissipation based on symmetric Reynolds stress tensors that can guarantee the conservation of the axial angular momentum in the control volumes (see Becker 2001). A good and comprehensive review of numerical dissipation methods in atmospheric models can be found in Jablonowski & Williamson (2011, p. 381).

The order of the hyperdiffusion operator is related to the number of times  $\nabla^2$  is applied to a prognostic variable. In the case of applying the operator once, we have a second-order diffusion. This low-order diffusion can be associated with, for example, heat diffusion or molecular diffusion, but since these are phenomena that work at much smaller scales than the ones resolved by any GCM, these low-order schemes should not be used. However, some models use second-order diffusion as numerical sponge schemes at the top of the model domain to effectively reduce the spurious reflections with the rigid lid at the top. In all the results explored with THOR, there was no indication that a sponge layer was needed, so we have not included it in the model. The use of any sponge layer in the model would also compromise the axial angular momentum conservation in the model top layers. The need to implement higher-order schemes in numerical models is related to the prediction from 2D turbulence theory, which tells us that in the infinite Reynolds number limit the energy dissipation rate is zero (Sadourny & Maynard 1995). This means that higher-order schemes are more scale-selective and help to maximize the ratio of enstrophy to energy dissipation. Increasing the scheme's order affects, however, the model's performance because of the increase in complexity of the equations by the recursive application of  $\nabla^2$ , and a compromise is made between performance and accuracy. Another potential problem with higher-order operators is related to the increase of the amplitudes of overshoots and undershoots with the increase of the operator order. In our work, we use fourth-order operators formulated in Cartesian coordinates that have the advantage of simplifying the equations since there is no need to

deal with metric terms. A misrepresentation of these geometric terms can also lead to problems in the axial angular momentum conservation.

The extra fluxes from the explicit fourth-order hyperdiffusion applied to the prognostic variables in this work have the following form:

$$F_\rho = -\nabla_h^2 K_d \nabla_h^2 \rho, \quad (43)$$

$$F_{v_{hx}} = -\nabla_h^2 K_d \nabla_h^2 v_{hx} - K_d \nabla_h^2 \nabla_h (\nabla_h \cdot (\rho \mathbf{v}_h)) + \frac{1}{r^2} \frac{\partial}{\partial r} (\rho v_r r^2), \quad (44)$$

$$F_{v_{hy}} = -\nabla_h^2 K_d \nabla_h^2 v_{hy} - K_d \nabla_h^2 \nabla_h (\nabla_h \cdot (\rho \mathbf{v}_h)) + \frac{1}{r^2} \frac{\partial}{\partial r} (\rho v_r r^2), \quad (45)$$

$$F_{v_{hz}} = -\nabla_h^2 K_d \nabla_h^2 v_{hz} - K_d \nabla_h^2 \nabla_h (\nabla_h \cdot (\rho \mathbf{v}_h)) + \frac{1}{r^2} \frac{\partial}{\partial r} (\rho v_r r^2), \quad (46)$$

$$F_{v_r} = -\nabla_h^2 \rho K_d \nabla_h^2 v_r, \quad (47)$$

$$F_p = -R_d \nabla_h^2 \rho K_d \nabla_h^2 T. \quad (48)$$

where  $F_\rho$  is the diffuse flux for density;  $F_{v_{hx}}$ ,  $F_{v_{hy}}$ ,  $F_{v_{hz}}$ , and  $F_{v_r}$  are the dissipative fluxes for the momentum projections; and  $F_p$  is the diffuse flux for the pressure field. This last term is applied after the pressure is updated.

The second term in Equations (44)–(46) is the 3D divergence damping, and it is used in our model with the main purpose of damping the high-frequency gravity noise mainly produced by the multiple-stage time integration explained in Section 3 (Skamarock & Klemp 1992). The strength of the 3D divergence damping was chosen in this work to be the same as the one used for the hyperdiffusion  $K_d$ . The divergence damping term is updated every short time step, while the other diffuse terms are updated at the large step (Tomita & Satoh 2004).

In order to improve the accuracy of the  $\nabla_h^2$  calculation, we first compute the gradient using Equation (7) in the primary grid (triangles), which places the calculated vectors at the corners of the control volume, avoiding an extra interpolation step. Then using these vectors we apply Equation (6) to the control volume to obtain  $\nabla_h^2$ .

The value of  $K_d$  once found for a set of particular planet conditions can be rescaled if the horizontal resolution or time is changed. For this rescaling method, we follow the work from Tomita & Satoh (2004) for a fourth-order scheme:

$$K_d = \mathcal{D} \frac{\bar{d}^4}{\Delta t}, \quad (49)$$

where  $\Delta t$  is the time step,  $\mathcal{D}$  is a nondimensional diffusion parameter, and  $\bar{d}$  is the average grid spacing defined in Equation (1). The damping timescale ( $\tau_d$ ) can be estimated using

$$\tau_d = \frac{\bar{d}^4}{2^5 K_d}. \quad (50)$$

Note that if any change is done to the space or time resolution, the strength of the diffusion may have to be adapted.

## 5. PROFX PHYSICS CORE

The physics core called PROFX contains the physical parameterizations that represent radiation, convective adjustments, and the mechanical interaction between surface and atmosphere. In this work we use highly simplified physics schemes, to simplify the interpretation of the results of the complex dynamical core. The prognostics variables updated in the physical core are computed using a backward Euler step from the products of the dynamical core (Ullrich & Jablonowski 2012). For the boundary layer, we use the following equation:

$$(\rho \mathbf{v}_h)^{\text{new}} = \frac{(\rho \mathbf{v}_h)^{\text{dyn}}}{1 + K_v (\sigma^{\text{dyn}}) \Delta t} \quad (51)$$

where  $(\rho \mathbf{v}_h)^{\text{new}}$  is the new updated variable from the physics scheme,  $\Delta t$  is the time step (the same as in the dynamical core for the large steps),  $(\rho \mathbf{v}_h)^{\text{dyn}}$  is the updated variable from the dynamical core, and  $K_v$  controls the strength of the damping. The variable  $\sigma$  is defined as the ratio between the pressure layer and the surface pressure. The surface pressure was estimated by extrapolating the pressures to the surface using the hydrostatic equation and keeping the temperature constant in that region. This equation forces the horizontal wind speed to zero inside the boundary layer region defined by  $K_v$ .

For the radiation/convection representation, the temperature is forced using

$$T^{\text{new}} = \frac{T^{\text{dyn}} + K_T (\sigma^{\text{dyn}}) \Delta t T_{\text{eq}}}{1 + K_T (\sigma^{\text{dyn}}) \Delta t} \quad (52)$$

In Equation (52),  $T^{\text{new}}$  and  $T^{\text{dyn}}$  are the updated temperatures from the physical and dynamical cores, respectively,  $K_T$  is the Newtonian cooling function toward radiative-convective equilibrium, and  $T_{\text{eq}}$  is the prescribed basic state temperature field. When the temperature at a particular location is larger than  $T_{\text{eq}}$ , the heating rate is negative ( $Q < 0$ ), and the opposite is true when the temperature is cooler than  $T_{\text{eq}}$ . In next section the form of the functions  $K_v$  and  $K_T$  in Equations (51) and (52) are defined for two different atmospheres.

## 6. GPU IMPLEMENTATION

We are currently analyzing the best strategies to boost the performance of THOR. Our existing program runs in a GPU environment. We integrate our simulations in an NVIDIA Tesla K20 GPU card, which has 2496 cores, has a memory bandwidth of 208 GB/s, supports double precision floating points, and includes error checking and correction (ECC) memory. ECC memory comprises extra bits of memory intended to detect and fix memory errors, which is important to ensure consistency between results computed at different times.

GPUs and central processing units (CPUs) have different architectures. The main advantage of using GPUs is the massively parallel architecture. In general, high-performance simulations in the GPU reach better power and cost efficiency than in the CPU. For this reason, GPUs have become attractive solutions for scientific computing applications (high performance at relatively low cost). Several Earth GCMs around the world are already being accelerated with GPU technology. Some examples are ASUCA (Shimokawabe et al. 2010),



**Table 1**  
Planet Parameters for the Test Cases

Parameters	Earth	Hot Jupiter
Planetary radius (m)	$6.371 \times 10^6$	$1.0 \times 10^8$
Planetary rotation rate ( $\text{s}^{-1}$ )	$7.292 \times 10^{-5}$	$2.1 \times 10^{-5}$
Surface gravity ( $\text{m s}^{-2}$ )	9.8	8.0
Specific heat ( $\text{J K}^{-1} \text{kg}^{-1}$ )	1004.6	13226.5
Gas constant ( $\text{J K}^{-1} \text{kg}^{-1}$ )	287.04	3779.0
Diffusion timescale (s)	$6.46 \times 10^3$	941.2

GEOS-5 (Putnam 2011), and WRF (Michalakes & Vachharajani 2008). In these examples, significant speedups of the simulations due to GPU integration have been reported (tens of times compared to simulations using the conventional CPU).

The programming language adopted in this work is CUDA-C. This language is essentially C/C++ with extra commands and functions that allow one, for example, to store data in the GPU memory and execute them in parallel calculations in the many threads available in the GPU. Codes in CUDA-C are programmed in both CPU and GPU, which are referred to as host and device, respectively. Our code is divided into two parts. In the first part, the model variables are declared, initialized, and allocated in the host and device memories. In this part, we also construct the model's grid and mathematical operators (e.g., divergence and gradient) and set the initial conditions of the atmosphere. At the end, we transfer the data from the host to the device memory. In the second part, we execute the multiple kernels (functions executed in the device) that solve the dynamical and physical core equations, and transfer the results back to the host. Since the data transfer between the host and the device through the peripheral component interconnect express is significantly expensive, we perform all the computations of the dynamical and physical cores in the GPU. Currently, the prognostic variables are transferred back to the CPU to be stored every 1000 time steps, which is a parameter that can be easily adapted. The kernels launched in the dynamical core execute individual threads that are grouped in 2D blocks of 256 threads. Within this compact block, the different threads synchronize and communicate via the fast on-chip shared memory. The block halo is accessed from a list of points stored in the global memory. To decompose the grid domain into subdomains (thread blocks), we take advantage of the geometrical properties of the icosahedron, which can be divided into 10 rhombuses. To optimize the simulations, by selecting the appropriate size of the blocks, we divide the main rhombuses into smaller subdomains (smaller rhombuses), as suggested in Randall et al. (2002).

## 7. BENCHMARKING

In this section, we explore two different atmospheres in two very distinct regimes: Earth (“Held–Suarez” test from Held & Suarez 1994) and a hot Jupiter (e.g., Menou & Rauscher 2009 and Heng et al. 2011b). The goal of these two tests is to assess the model performance for long integrations under different forcings and analyze the main statistical dynamical quantities of the atmosphere. The Held–Suarez test from Held & Suarez (1994) is often used in the Earth climate community as a benchmark test of dynamical cores, and the hot Jupiter test case has also been explored by different models in the astrophysics community (e.g., Menou & Rauscher 2009; Heng et al. 2011b;

Bending et al. 2013; Mayne et al. 2014). The vast literature on these tests is essential in the validation and analysis of our methods to solve the Euler equations. These tests are not aimed at reproducing a realistic atmospheric circulation since they use very simple representations of adiabatic heating and dissipative processes. However, the simple parameterizations force the model to a climate state qualitatively similar to what is expected for each planet. It also allows for an easier study of the nonlinear processes at work in the atmospheric circulation, to verify if the main atmospheric circulation drivers are well represented and check if any atmospheric phenomena are produced for unphysical reasons.

The simulations explored in this work were integrated over a period of 1200 Earth days. In both simulations, the first 200 days were discarded from the analysis, since the data in this period is regarded as part of the spin-up phase of the model. The number of grid cells is also the same in the two experiments. In the construction of the icosahedral grid, we apply five times the recursive method, obtaining a grid g-level 5 that has 10,242 points quasi-uniformly distributed horizontally ( $\approx 2^\circ$  resolution), and for each point we use 37 vertical layers.

The results presented in the next section use pressure as the vertical axis, which is done to better compare our results with other groups that presented the results in the same format. This step is not done during the model integration but as a postprocessing step where the values are interpolated linearly to a pressure grid.

We do not explore the case of a deep hot Jupiter test as shown in Heng et al. (2011b) because the vertical domain of the physical scheme that forces the temperature is not suitable for the vertical coordinate system used in THOR. In THOR the vertical domain is defined in altitude, and in Heng et al. (2011b) it is defined in pressure. Note that this is a problem related to the formulation of the experiment and not related to the ability of THOR to simulate deep atmospheres. Using the equations presented in Heng et al. (2011b), we found that at the highest layers the heating rates were unrealistically large, due to the low pressures that were being obtained outside the pressure range defined for the test. Mayne et al. (2014) use a model with the same vertical axis as in THOR, and they correct the heating rates for the lowest pressure to avoid the numerical problems and to be able to simulate the deep atmospheric test case. We defer such work to a future paper.

### 7.1. Benchmark Test for Earth

We start the validation of the model by exploring the Held–Suarez test (Held & Suarez 1994). In this test, the adiabatic heating and frictional forcing are parameterized to represent an atmospheric circulation qualitatively similar to Earth. The two parameterizations are based on two linear forcing schemes represented by Equations (51) and (52). In Equation (51) the strength of the frictional damping is defined by

$$K_v(\sigma) = k_f \times \max\left(0.0, \frac{\sigma - \sigma_b}{1 - \sigma_b}\right), \quad (53)$$

where  $k_f^{-1} = 1$  Earth day, and  $\sigma_b = 0.7$ . The functions  $T_{\text{eq}}$  and  $K_T$  from the adiabatic heating function (Equation (52)) are

given by

$$T_{\text{eq}} = \max \{200 \text{ K}, [315 \text{ K} - (\Delta T)_y \sin^2 \phi - (\Delta \theta)_z \log \left( \frac{p}{p^*} \right) \cos^2 \phi] \left( \frac{p}{p^*} \right)^k \}, \quad (54)$$

$$K_T = k_a + (k_s - k_a) \times \left( 0, \frac{\sigma - \sigma_b}{1 - \sigma_b} \right) \cos^4 \phi. \quad (55)$$

In the above equations,  $k_a^{-1} = 40$  Earth days,  $k_s^{-1} = 4$  Earth days,  $(\Delta T) = 60 \text{ K}$ ,  $k = R_d/C_p$ , and  $\phi$  is the latitude. The structure of the function  $T_{\text{eq}}$  represents a stable stratification at low latitudes, which then decreases to zero poleward. This poleward decrease reduces the excitation of gravitational instabilities (Held & Suarez 1994).

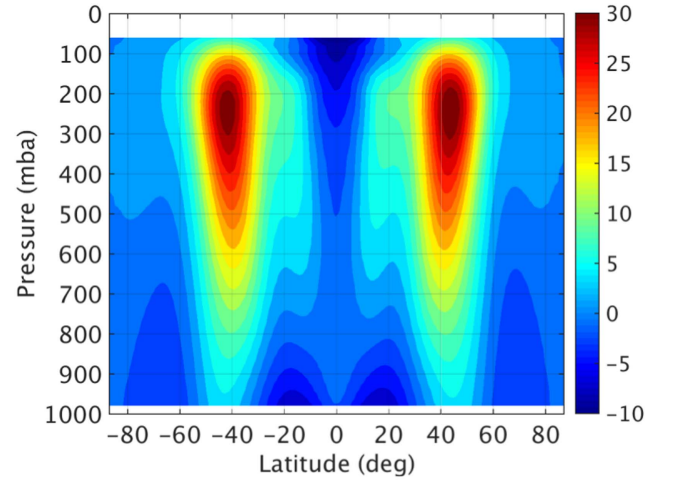
This simulation started from a rest atmosphere and with the same temperature profile in every column taken from the function  $T_{\text{eq}}$  at the equator in hydrostatic equilibrium. Starting the simulation from a nonisothermal atmosphere helps to break the symmetry between the two hemispheres. This problem is related to the existence of an unstable symmetric solution for the Held–Suarez experiment. A sudden breaking of the symmetry during the simulation can lead to the formation of large instabilities, which can lead to unrealistic values. There are other techniques to break the symmetry, such as an isothermal atmosphere with small perturbations (Held & Suarez 1994) or initial perturbations in the vorticity field (Heng et al. 2011b).

The simulation used a time step of 1000 s, a grid refinement equal to g-level 5, 37 vertical layers covering from 0 to 32 km, and a diffusion timescale of  $6.46 \times 10^3 \text{ s}$ . The simulation has 200 Earth days of a spin-up phase where the data results are discarded. During this time, the available potential energy stored in the atmosphere is converted to kinetic energy. This conversion is initially forced by the temperature forcing and instabilities induced by the planet’s rotation. Before the end of the spin-up phase time, the atmospheric circulation has reached a statistical equilibrium state, where for example the total axial angular momentum has a statistically steady average value as a function of time.

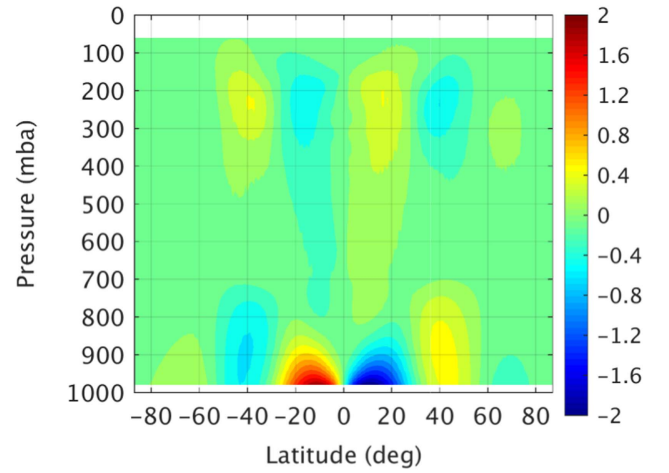
In Figure 7, we show the zonal and meridional wind and the mass stream function for the Held–Suarez experiment. The mass stream function ( $\Psi$ ) is calculated using the expression

$$\Psi = \frac{2\pi A \cos \phi}{g} \int_0^p [\bar{v}] dp. \quad (56)$$

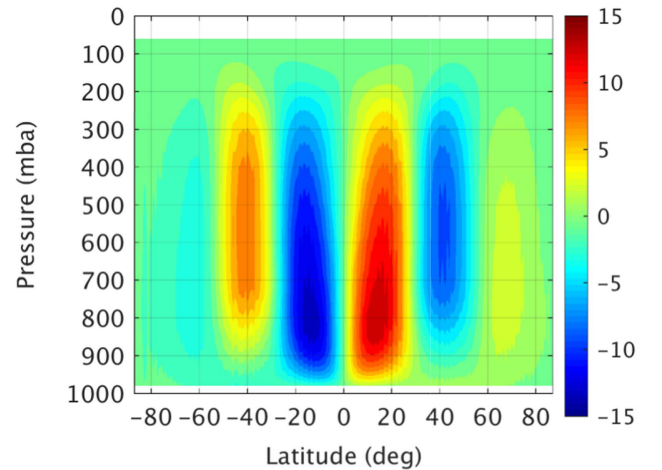
In this equation,  $A$  represents the planet radius,  $p$  is the pressure, and  $[\bar{v}]$  is the zonal and time-averaged meridional wind. The simulation produces one jet in each hemisphere’s center at roughly 250 mbar and  $45^\circ$  latitude consistent with the results from Held & Suarez (1994). The zonal wind structure matches closely the results obtained in Held & Suarez (1994). The zonal wind changes sign in some regions in the atmosphere: at low latitudes it appears in the upper atmosphere, and in the lower atmosphere it appears at low latitudes and also at latitudes higher than  $60^\circ$ . The meridional component of the wind and the mass stream function show the classical atmospheric structure in the upper atmosphere: the Hadley cells at low latitudes, followed by the Ferrel cells that extend to  $60^\circ$  latitude, and then the polar cells (see also Heng



(a) Eastward wind speed



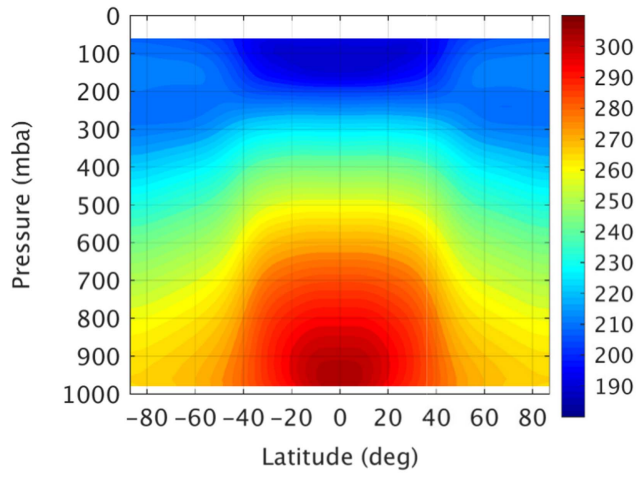
(b) Northward wind speed



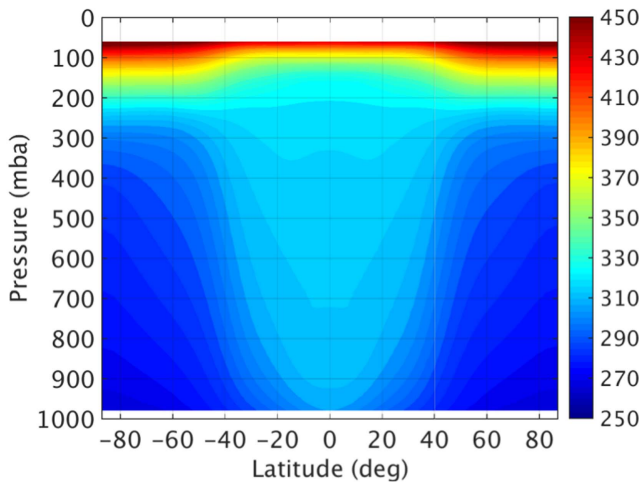
(c) Mass stream function

**Figure 7.** Final averaged zonal and meridional winds ( $\text{m s}^{-1}$ ) and mass stream function ( $10^{10} \text{ kg s}^{-1}$ ) for the Held–Suarez experiment. The values were time averaged for 1000 Earth days.

et al. 2011a for a comparison). In the lower atmosphere, the meridional winds change direction and become slightly stronger.



(a) Temperature



(b) Potential temperature

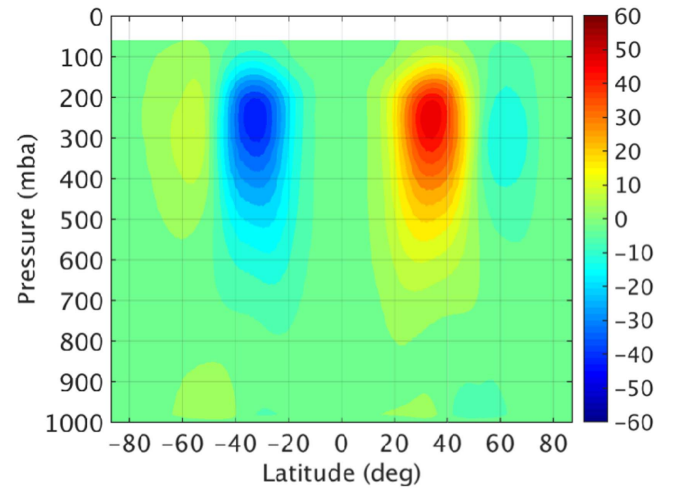
**Figure 8.** Final averaged temperature and potential temperature (K) for the Held-Suarez experiment. The values were time averaged for 1000 Earth days.

The averaged temperature and potential temperatures are shown in Figure 8. These two averaged maps are very similar to the ones shown in Held & Suarez (1994). The resulting simulated potential temperature shows a weaker vertical gradient in the tropics than at higher latitudes. This configuration indicates weaker static stability at lower latitudes, which drives more upward motion in that region. The cold stratospheric region is obtained because of the constant basic state temperature in the temperature-forcing scheme, which makes that region of the atmosphere well stratified and inactive.

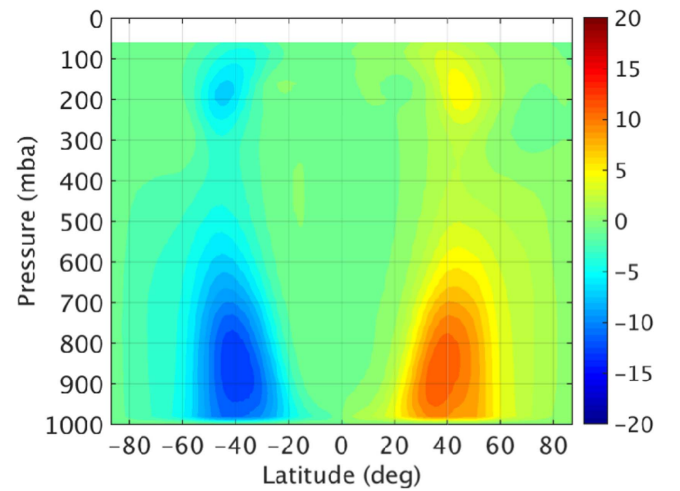
Figures 9 and 10 are related to wave activity present in the simulation. This activity has a crucial role in the atmospheric circulation produced, and the results are validated by comparing them to results from Held & Suarez (1994), Jablonowski (1998), and Ullrich & Jablonowski (2012). The zonal inhomogeneities or eddy components of the variables are represented with the dash symbol:

$$\phi' = \phi - \bar{\phi}, \quad (57)$$

where  $\bar{\phi}$  represents the zonal averaged quantity of  $\phi$ . Figure 9(a) shows the meridional eddy flux of the zonal momentum. This flux has an important weight in the local



(a) Meridional eddy flux of zonal momentum



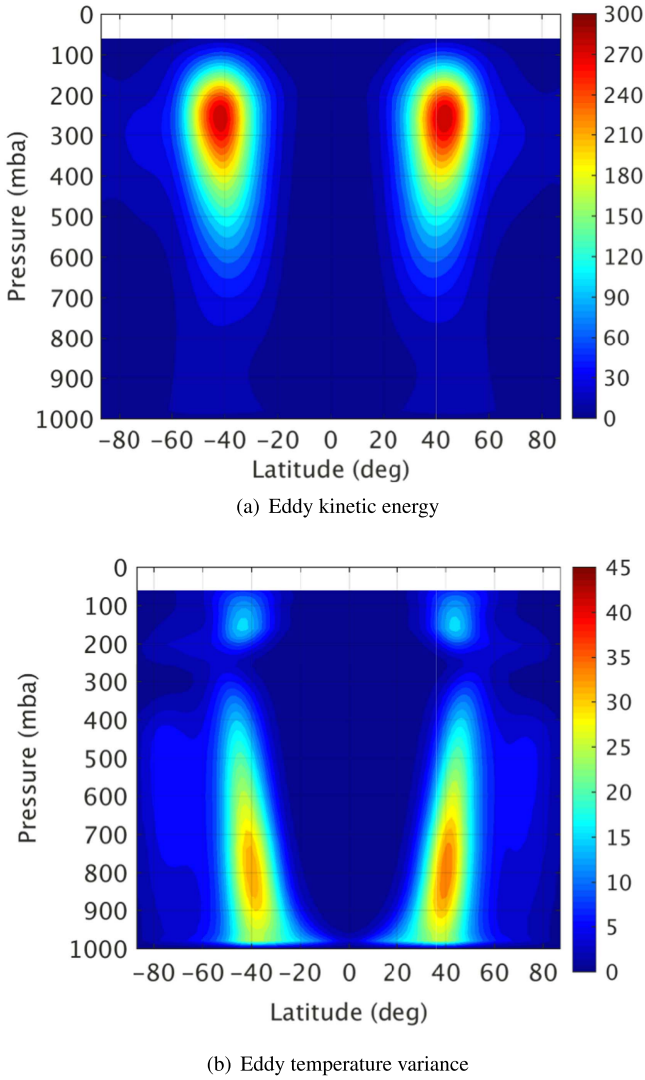
(b) Meridional eddy flux of temperature

**Figure 9.** Final averaged meridional eddy flux of zonal momentum ( $\text{m}^2 \text{s}^{-1}$ ) and temperature ( $\text{m s}^{-1} \text{K}$ ) for the Held-Suarez experiment. The values were zonal and time averaged for 1000 Earth days.

balance of the momentum and in the global circulation produced. The flux is more intense at pressure levels above 800 mba, and in general the zonal motion is transported from low latitudes toward midlatitudes. The divergence of the meridional eddy flux at low latitudes results in a persistent presence of a retrograde flow in that region. In the polar regions, waves transport zonal motions toward low latitudes that converge at midlatitudes. This convergence combined with the mean circulation form the prograde winds at midlatitude seen in Figure 7(a), with one local maximum in each hemisphere at roughly 250 mba. The eddy heat flux is shown in Figure 9(b). The heat is being transported by atmospheric waves in general from the equatorial regions toward higher latitudes. This is an important quantity for the energy balance of the atmosphere. The heat transport is strongest in the lower atmosphere, where the largest baroclinic wave activity is present.

The eddy kinetic energy and eddy temperature variance quantify the magnitude of the wave activity in the momentum and temperature fields. As expected from Figures 9(a) and (b),



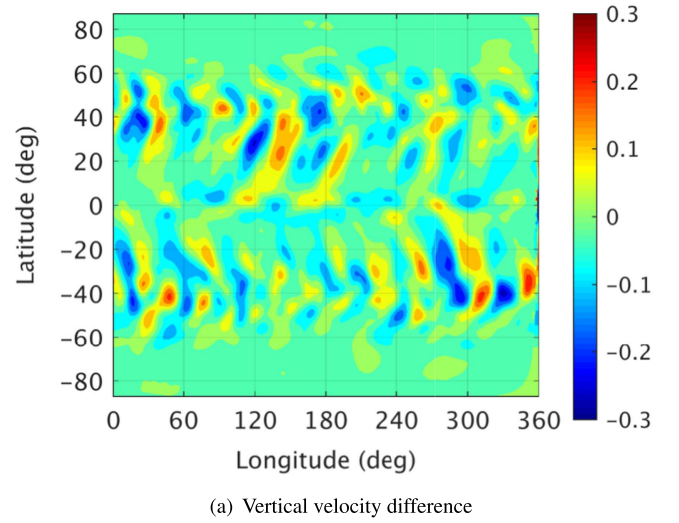


**Figure 10.** Final averaged eddy kinetic energy ( $\text{m}^2 \text{s}^{-2}$ ) and temperature variance ( $\text{K}^2$ ) for the Held–Suarez experiment. The values were time averaged for 1000 Earth days.

the largest amplitudes are located at midlatitudes. In the eddy kinetic energy case, Figure 10(a), the maxima are located in the zonal wind jet region. In Figure 10(b), the maxima in the temperature variance correspond to the maxima in the baroclinic wave activity in the lower atmosphere seen in Figure 9(a).

Comparing our results on the mean circulation and wave activity with the results from Held & Suarez (1994), we concluded that both results are quantitatively very similar. The similarity between the two models’ results demonstrate that our new model has passed this important benchmark test.

The geometrical grid properties can be associated with sources of numerical noise visible in the solution (see Section 2.2). In this work, this unphysical quantity, called grid imprinting, is reduced from the numerical solution when we applied the methods that smoothed the distortions in the standard icosahedral grid (“spring dynamics method”) and moved the points to the centroids of the control volumes. To verify the impact of the grid imprinting in our simulation, we



**Figure 11.** Final time-averaged vertical velocity difference ( $10^4 \text{ m s}^{-1}$ ) in the first model layer for the Held–Suarez experiment. The values were time averaged for 1000 Earth days.

analyzed a map of the vertical velocity difference ( $[w^*]$ ) averaged over the last 1000 Earth days of the long simulation (Figure 11). Here,  $w^*$  is defined as

$$w^* = w - [w]. \quad (58)$$

The square brackets represent the time-averaging operator. This quantity in the first layer is a good diagnostic for grid imprinting because it is much smaller than the other prognostic variables, and in the case of the presence of persistent, small perturbations from the grid structure, they would be easily seen from the  $[w^*]$  results (Ullrich & Jablonowski 2012). From Figure 11 we do not see any artificial values from the grid structure, which is an indication that the grid imprinting noise is efficiently reduced in our simulations.

## 7.2. Benchmark Test for Hot Jupiters

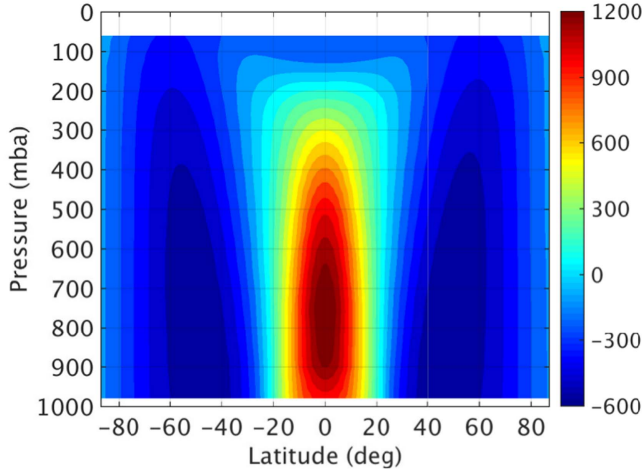
The new model is aimed at probing a large range of planetary conditions. Here we explore a simulation under the conditions of a typically tidally locked hot Jupiter proposed by Menou & Rauscher (2009). This experiment was suggested in Heng et al. (2011b) as a benchmark test, and it was also explored later in Bending et al. (2013) and Mayne et al. (2014).

This experiment was integrated without boundary layer friction at the bottom, as suggested in Menou & Rauscher (2009), which means that the function  $K_v$  is set to zero (no interchanges of momentum between the simulated atmosphere and the interior). The temperature equilibrium function ( $T_{\text{eq}}$ ) used in Equation (52) is defined for this experiment as

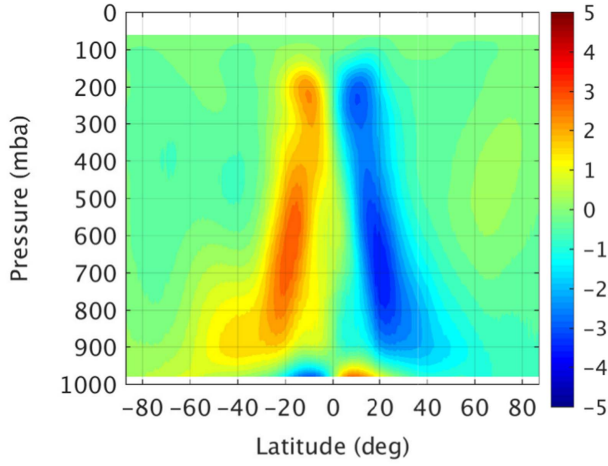
$$T_{\text{eq}} = T_{\text{vert}} + \beta_{\text{trop}} \Delta T_{E-P} \cos(\lambda) \cos(\phi), \quad (59)$$

where  $\Delta T_{E-P} = 300 \text{ K}$  (equator-to-pole difference),  $\lambda$  is the longitude,  $\phi$  is the latitude, and  $\beta_{\text{trop}}$  is defined by

$$\beta_{\text{trop}} = \begin{cases} \sin \frac{\pi(\sigma - \sigma_{\text{stra}})}{2(1 - \sigma_{\text{stra}})}, & z \leq z_{\text{stra}} \\ 0, & z > z_{\text{stra}} \end{cases} \quad (60)$$



(a) Eastward wind speed



(b) Mass stream function

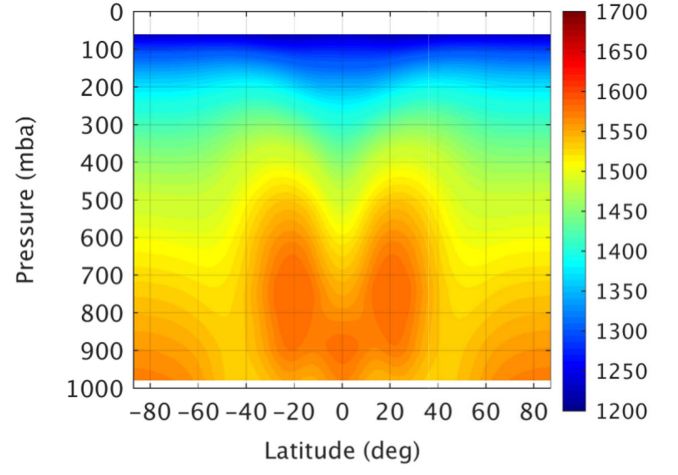
**Figure 12.** Final averaged zonal winds ( $\text{m s}^{-1}$ ) and mass stream function ( $10^{13} \text{ kg s}^{-1}$ ) for the hot Jupiter experiment. The values were time averaged for 1000 Earth days.

In Equation (60),  $\sigma_{\text{stra}} = 0.12$  and  $z_{\text{stra}} = 2 \times 10^3 \text{ km}$ . The function  $T_{\text{vert}}$  used in Equation (59) has the following form:

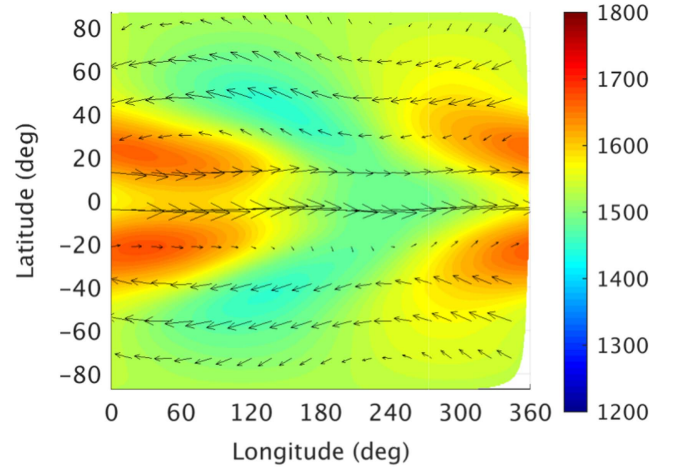
$$T_{\text{vert}} = \begin{cases} T_{\text{surf}} - \Gamma_{\text{trop}} \left( z_{\text{stra}} + \frac{z - z_{\text{stra}}}{2} \right) \\ + \left( \left[ \frac{\Gamma_{\text{trop}}(z - z_{\text{stra}})}{2} \right]^2 + \Delta T_{\text{stra}}^2 \right)^{1/2}, & z \leq z_{\text{stra}}, \\ T_{\text{surf}} - \Gamma_{\text{trop}} z_{\text{stra}} + \Delta T_{\text{stra}}, & z > z_{\text{stra}} \end{cases} \quad (61)$$

where  $T_{\text{surf}} = 1600 \text{ K}$ ,  $\Gamma_{\text{trop}} = 2 \times 10^{-4} \text{ K m}^{-1}$ , and  $\Delta T_{\text{stra}} = 10 \text{ K}$ . The timescale in the temperature-forcing scheme is set to a constant value equal to  $1.5 \times 10^5 \text{ s}$ . The initial conditions in this experiment were an isothermal atmosphere at  $1600 \text{ K}$  in hydrostatic equilibrium and the wind velocity components set to zero. The simulation used a time step of  $600 \text{ s}$ , a grid refinement equal to g-level 5, 37 vertical layers covering from 0 to  $4875 \text{ km}$  altitude, and a diffusion timescale of  $941.2 \text{ s}$ .

The atmospheric circulation produced is very distinct from the previous Earth-like case. Here the atmospheric circulation is largely influenced by the strong day–night contrast in the



(a) Temperature



(b) Temperature at sigma level 0.7

**Figure 13.** (a) Final averaged temperature (K) for the hot Jupiter experiment. (b) Final time-averaged horizontal map of temperature at sigma level 0.7. The arrows show the time-averaged direction of the wind speed. In both maps the values were averaged for 1000 Earth days.

temperature forcing. Figure 12(a) shows the zonal winds averaged in longitude and in time over the last 1000 Earth days of the simulation. In general, the zonal winds show a strong prograde jet at low latitudes and centered at a pressure level of  $800 \text{ mbar}$ . The mass stream function shows the presence of indirect atmospheric cells (eddy-driven cells) at low latitudes with two peaks in the mass stream function in each hemisphere at different pressure levels: roughly  $800$  and  $250 \text{ mbar}$ . This mean circulation combined with the excited equatorial waves transport prograde axial angular momentum toward low latitudes and help maintain the strong prograde equatorial jet. At higher latitudes, the direction of the winds reverse. These retrograde winds are expected from the conservation of the total axial angular momentum in the atmosphere. In these regions, the mass stream function map shows that the flow follows on average a clockwise direction in the northern hemisphere and an anticlockwise direction in the southern hemisphere.

The averaged temperature map is shown in Figure 13(a). In this map, we see clearly two maxima in the temperature in each hemisphere, which are related to the efficient heat transport

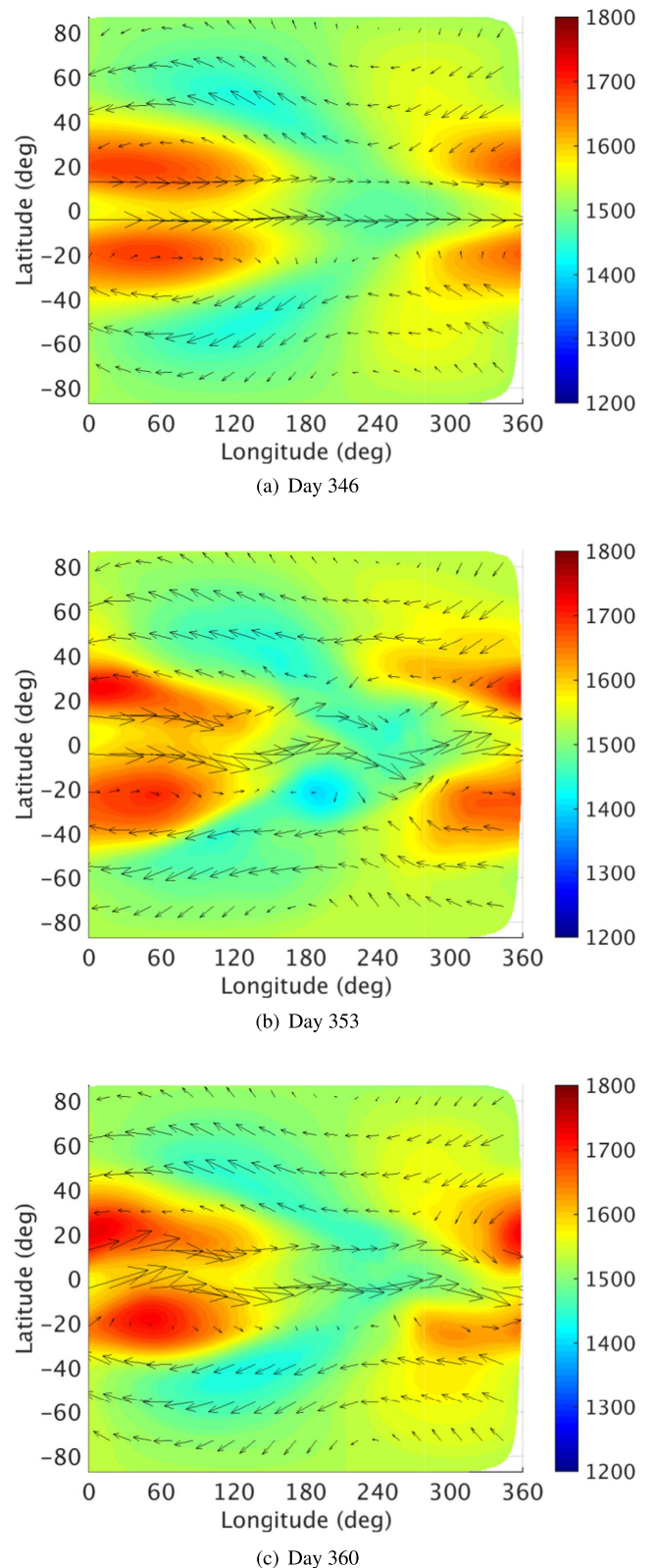
in the equatorial region. Figure 13(b) shows a time-averaged map of the temperature and winds at the pressure level 700 mbar. The two maxima in the temperature in each hemisphere are clear in this map as well. The heat is being transported in opposite directions at low (prograde transport) and high latitudes (retrograde transport). At low latitudes, it is efficiently transported by the broad, strong equatorial jet. Figure 14 shows snapshots of the temperatures and winds at sigma level 0.7. The map at day 246 has been used in Heng et al. (2011b), Bending et al. (2013), and Mayne et al. (2014) for comparison with the work of Menou & Rauscher (2009). However, one should be careful when comparing instantaneous results because of the variability of the flow.

In general, the results obtained with our model are quantitatively very similar to the ones obtained by Menou & Rauscher (2009), Heng et al. (2011b), and Bending et al. (2013). Note that the substellar point in our model is shifted by  $\pi$  when compared with the Menou & Rauscher (2009) and Heng et al. (2011b) results. The results obtained in Mayne et al. (2014) are qualitatively similar to the other models, but quantitatively very different. The main differences are at the tropics. The magnitude of the equatorial jet is 40% weaker and is located much lower in pressure (400 mbar). These differences in the results have an impact on the heat transport of the simulated atmosphere, as is clear when comparing their averaged temperature field results with the other models (including our model). In Mayne et al. (2014), the differences in the model domain and boundary conditions applied are suggested as possible causes for the differences in the results. The differences in the domain are attributed to the use of different vertical coordinates between the models: pressure coordinates in Menou & Rauscher (2009) and Heng et al. (2011b), and height in Mayne et al. (2014). However, in our model we use the same model domain and boundaries as in Mayne et al. (2014), and our results are very similar to the Menou & Rauscher (2009) and Heng et al. (2011b) results. For this reason, the cause for the disagreement in the results between Mayne et al. (2014) and the other models remains unclear.

## 8. CONCLUSION

We have developed a new GCM for a large diversity of atmospheric conditions. This new platform has passed two important benchmark tests with very different atmospheric conditions: for Earth and for a hot Jupiter-like planet. The grid implemented is able to reach a second-order accuracy, and the grid imprinting noise is efficiently removed from the simulations. In this, work we have described the structure of a new complex dynamical model that uses a numerical solver based on an explicit-implicit formulation. This formulation allows for a very good performance since we use a split-explicit time-step method coupled with an implicit integration of the vertical momentum, which is numerically constrained by the propagation of the fast sound waves across the fine vertical resolution. The model does not use any of the traditional approximations often used in Earth climate studies that can limit the model flexibility in exploring other planetary conditions. This new platform solves the deep nonhydrostatic Euler equations, and more physical representations can be easily added to the original set of equations.

The two experiments that were successfully done represent a small sample in the general diversity of planets. However,



**Figure 14.** Snapshots of the temperature field at sigma 0.7 for the days 346, 353, and 360. Menou & Rauscher (2009) and Heng et al. (2011b) only show day 346. The arrows show the direction of the circulation.

THOR showed a very good performance in the simulation of these two very distinct atmospheric conditions, which allows us to be in an advantageous position to continue exploring the



diversity of planetary parameters that are expected to characterize the atmospheres of solar system and extrasolar planets.

In summary, the main advantages of using our new platform over other recent planetary models are as follows:

1. The resolved atmospheric fluid flow is completely represented, and no approximations are used that could compromise the physics of the problem.
2. The model uses, for the first time in exoplanetary studies, an icosahedral grid that solves the pole problem.
3. The interface is user friendly and can be easily adapted to a multitude of atmospheric conditions.

We have developed a solid basis for a virtual atmospheric laboratory. This model has also been developed to keep the user interface easy to use since we aim to make this code free and open source to the community (see more details about THOR at [www.exoclimate.net](http://www.exoclimate.net)).

J.M.M., S.G., L.G., and K.H. acknowledge financial and administrative support from the Center for Space and Habitability (CSH), the PlanetS National Centre of Competence in Research (NCCR), the Swiss National Science Foundation (SNF), and the Swiss-based MERAC Foundation. THOR is part of the Exoclim Simulation Platform (ESP) that will be freely available to the community (see [www.exoclims.net](http://www.exoclims.net)).

## REFERENCES

- Arakawa, A., & Konor, C. S. 2009, *MWRv*, **137**, 710
- Baumgardner, J. R., & Frederickson, P. O. 1985, *SJNA*, **22**, 1107
- Becker, E. 2001, *JAtS*, **58**, 269
- Bending, V. L., Lewis, S. R., & Kolb, U. 2013, *MNRAS*, **428**, 2874
- Boville, B. A., & Gent, P. R. 1998, *JCLI*, **11**, 1115
- Dowling, T. E. 2013, in *Earth General Circulation Models*, ed. S. J. Mackwell et al. (Tucson, AZ: Univ. Arizona Press), 193
- Dowling, T. E., Fischer, A. S., Gierasch, P. J., et al. 1998, *Icar*, **132**, 221
- Dubos, T., & Voitus, F. 2014, *JAtS*, **71**, 4621
- Durran, D. R., & Blossey, P. N. 2012, *MWRv*, **140**, 1307
- Forget, F., Hourdin, F., Fournier, R., et al. 1999, *JGR*, **104**, 24155
- Grimm, S. L., & Heng, K. 2015, *ApJ*, **808**, 182
- Heikes, R., & Randall, D. A. 1995, *MWRv*, **123**, 1862
- Heikes, R. P., Randall, D. A., & Konor, C. S. 2013, *MWRv*, **141**, 4450
- Held, I. M., & Suarez, M. J. 1994, *BAMS*, **75**, 1825
- Heng, K., Frierson, D. M. W., & Philipps, P. J. 2011a, *MNRAS*, **418**, 2669
- Heng, K., Menou, K., & Philipps, P. J. 2011b, *MNRAS*, **413**, 2380
- Heng, K., & Showman, A. P. 2015, *AREPS*, **43**, 509
- Hourdin, F., Talagrand, O., Sadourny, R., et al. 1995, *Icar*, **117**, 358
- Jablonowski, C. 1998, PhD thesis, Univ. Bonn
- Jablonowski, C., & Williamson, D. 2011, *Numerical Techniques for Global Atmospheric Models* (Berlin: Springer)
- Kataria, T., Showman, A. P., Fortney, J. J., et al. 2015, *ApJ*, **801**, 86
- Klemp, J. B., Skamarock, W. C., & Dudhia, J. 2007, *MWRv*, **135**, 2897
- Knutson, H. A., Charbonneau, D., Cowan, N. B., et al. 2009, *ApJ*, **690**, 822
- Lebonnois, S., Hourdin, F., Eymet, V., et al. 2010, *JGRE*, **115**, E06006
- LeVeque, R. 2002, *Finite Volume Methods for Hyperbolic Problems* (Cambridge: Cambridge Univ. Press)
- Majewski, D., Liermann, D., Prohl, P., et al. 2002, *MWRv*, **130**, 319
- Mayne, N. J., Baraffe, I., Acreman, D. M., et al. 2014, *A&A*, **561**, A1
- Mendonça, J., & Read, P. 2016, *P&SS*, submitted
- Menou, K., & Rauscher, E. 2009, *ApJ*, **700**, 887
- Michalakes, J., & Vachharajani, M. 2008, *Parallel Process. Lett.*, **18**, 531
- Ogura, Y., & Phillips, N. A. 1962, *JAtS*, **19**, 173
- Ooyama, K. V. 1990, *JAtS*, **47**, 2580
- Phillips, N. A. 1966, *JAtS*, **23**, 626
- Putnam, W. 2011, NASA Tech. Rep. (Greenbelt, MD: Goddard Space Flight Center), <http://ntrs.nasa.gov/search.jsp?R=20120009084>
- Randall, D., Wood, R., Bony, S., et al. 2007, in *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*, ed. S. D. Solomon et al. (Cambridge: Cambridge Univ. Press), 589
- Randall, D. A., Ringler, T. D., Heikes, R. P., Jones, P., & Baumgardner, J. 2002, *CSE*, **4**, 32
- Rauscher, E., & Menou, K. 2012, *ApJ*, **745**, 78
- Ringler, T. D., & Randall, D. A. 2002, *MWRv*, **130**, 1411
- Rogers, T. M., & Showman, A. P. 2014, *ApJL*, **782**, L4
- Sadourny, R., Arakawa, A., & Mintz, Y. 1968, *MWRv*, **96**, 351
- Sadourny, R., & Maynard, K. 1995, *Atmos. Ocean*, **35**, 347
- Sato, M. 2002, *MWRv*, **130**, 1227
- Schneider, T., & Liu, J. 2009, *JAtS*, **66**, 579
- Shimokawabe, T., Aoki, T., Muroi, C., et al. 2010, in *SC Conference* (New York: IEEE), 1
- Showman, A. P., Fortney, J. J., Lian, Y., et al. 2009, *ApJ*, **699**, 564
- Showman, A. P., & Guillot, T. 2002, *A&A*, **385**, 166
- Shutts, G. 2005, *QJRM*, **131**, 3079
- Skamarock, W. C., & Klemp, J. B. 1992, *MWRv*, **120**, 2109
- Skamarock, W. C., & Klemp, J. B. 2008, *JCoPh*, **227**, 3465
- Smagorinsky, J. 1963, *MWRv*, **91**, 99
- Staniforth, A. 2014, *QJRM*, **140**, 2685
- Staniforth, A., & Thuburn, J. 2012, *QJRM*, **138**, 1
- Staniforth, A., Wood, N., & Girard, C. 2003, *QJRM*, **129**, 3495
- Stuhne, G. R., & Peltier, W. R. 1996, *JCoPh*, **128**, 58
- Thrastarson, H. T., & Cho, J. Y. 2011, *ApJ*, **729**, 117
- Thuburn, J. 2008, *JCoPh*, **227**, 3715
- Tomita, H., & Sato, M. 2004, *FDyR*, **34**, 357
- Tomita, H., Tsugawa, M., Sato, M., & Goto, K. 2001, *JCoPh*, **174**, 579
- Ullrich, P. A., & Jablonowski, C. 2012, *JCoPh*, **231**, 5078
- Wang, N., & Lee, J. 2011, *SIAM J. Sci. Comput.*, **33**, 2536
- White, A. A., Hoskins, B. J., Roulstone, I., & Staniforth, A. 2005, *QJRM*, **131**, 2081
- Wicker, L. J., & Skamarock, W. C. 2002, *MWRv*, **130**, 2088
- Williamson, D. L. 1968, *Tell*, **20**, 642
- Williamson, D. L., Olson, J. G., & Jablonowski, C. 2009, *MWRv*, **137**, 790
- Yamazaki, Y. H., Skeet, D. R., & Read, P. L. 2004, *P&SS*, **52**, 423
- Zängl, G., Reinert, D., Rípodas, P., & Baldauf, M. 2015, *QJRM*, **141**, 563